ON USE OF IDENTITY-BASED ENCRYPTION FOR SECURE EMAILING

Christian Veigner, Chunming Rong
University of Stavanger, 4036 Stavanger, Norway

Keywords: Identity-based encryption (IBE), public-key infrastructure (PKI), spam, viruses, denial of service attacks.

Abstract: In 1984 Adi Shamir requested a solution for a novel public-key encryption scheme, called identity-based encryption. The original motivation for identity-based encryption was to help the deployment of a public-key infrastructure. The idea of an identity-based encryption scheme is that the public key can be any arbitrary string, for example, an email address, a name or a role. Several solutions were proposed in the following years. In 2001 the first practical and efficient scheme was proposed by Boneh and Franklin. Their encryption scheme was based on the Weil pairing on elliptic curves and proved secure in the random oracle model. In 2005, a new promising suggestion due to Waters was proposed, this time as an efficient solution without random oracles. An identity-based encryption (IBE) scheme does not need to download certificates to authenticate public keys as in a public-key infrastructure (PKI). A public key in an identity-based cryptosystem is simply the receiver's identity, e.g. an email address. As often, when new technology occurs, the focus is on the functionality of the technology and not on its security. In this paper we briefly review about identity-based encryption and decryption, particularly, the Boneh-Franklin algorithms. Later on we show that IBE schemes used for secure emailing render spamming far easier for spammers compared to if a PKI certificate approach is used. With the IBE approach, viruses may also be spread out more efficiently.

1 INTRODUCTION

Recently, identity-based cryptography, i.e. identity-based encryption (IBE) and identity-based signatures (IBS), has been a popular topic of research in several research communities around the globe. Especially since Boneh and Franklin suggested the first practical and efficient identity-based encryption scheme from the Weil pairing on elliptic curves (Boneh, 2001). This solution came after several not-fully satisfactory proposals (Fiat, 1986, Feige, 1988). Some previous solutions required users not to collude, others that the Private Key Generator (PKG) spent a long time for each private key generation request. Some solutions even required tamper resistant hardware. It is fair to say that, until now, constructing a usable IBE system has been an open problem. In the same paper Boneh and Franklin also showed how an IBE scheme immediately could be converted into a signature scheme. Use of IBE was now suggested by different research communities for many different purposes (Boyen, 2003, Chen, 2002, Lynn, 2002, Waters, 2004). In (Veigner, 2006) we analyze the possibilities of using IBE for symmetric key agreement in Wireless Sensor Networks (WSN).

Use of IBE for email content encryption has also been suggested. Voltage (Voltage, 2004) offers secure business communication via email and instant massaging with end-to-end content level encryption through their SecureMail implementation. Their solution offers the first secure email solution that makes secure ad-hoc business communication as easy as traditional, non-encrypted messaging. The use of IBE in email systems opens a number of business opportunities not possible before; for example, external broker communication can now be conducted securely via email in a natural ad-hoc fashion.

Due to security concerns regarding privacy when sending emails, an ever increasing number of users apply encryption to their email content. The difficulties of symmetric keys scalability has lead to a world-wide acceptance of asymmetric cryptography for realizing privacy in such schemes. Nowadays, the award winning identity-based encryption (IBE) scheme designed by Boneh and Franklin (Boneh, 2001) is considered applicable in

Veigner C. and Rong C. (2006).
ON USE OF IDENTITY-BASED ENCRYPTION FOR SECURE EMAILING.
In *Proceedings of the International Conference on Security and Cryptography*, pages 289-296
DOI: 10.5220/0002099502890296

almost every cryptographic area. As mentioned, Voltage (Voltage, 2004) has implemented an IBE-suite for secure emailing. However due to certain properties of IBE, we fear that spammers now get an advantage in compare to when emails are encrypted using a PKI like scheme.

Spam, also referred to as unsolicited email, is often offensive and illegal. ISPs are strongly against it because it consumes the ISP's resources due to its vast volumes and angers the ISP's customers. Most spam mails are meant to promote a product or service. However, very few of these emails will include a valid *from:* address, so tracing their origin can be challenging.

In IBE there is no need for sender Alice to obtain receiver Bob's public-key certificate. When Bob receives an encrypted email, he contacts a third party called Private Key Generator (PKG). Bob obtains his private key by authenticating himself to the PKG, in the same way he would authenticate himself to a CA in a PKI scheme. Bob can then read his email. Note that unlike the existing secure email infrastructure, Alice can send encrypted emails to Bob even when Bob has not yet set up his public-key certificate.

The rest of this paper is organized as follows. Section 2 introduces some available solutions on how to prevent spam when encryption is not applied. Section 3 introduces symmetric an asymmetric cryptography used in email solutions. Section 4 describes basic ideas and properties of identity-based encryption, particularly, the Boneh-Franklin scheme. In section 5, we give a brief intro on how IBE may be used for securing emails, followed by section 6 which concludes this paper.

2 PREVENTING EMAIL-SPAM AND VIRUSES IN UNENCRYPTED EMAILS

Generally, there are two different angles of incidence for a spammer to dispatch unsolicited emails. Originally, spammers used their own servers to generate and send spam and thereby devised techniques to avoid being blacklisted. Today, spammers often rely on virus writers and hackers to provide a constant supply of servers to hide their identity and generate huge volumes of mail. We will show that protection against both of these techniques is already available.

2.1 Filtering Incoming Emails

Several companies offer email-filtering technology to customers, e.g. (Securence, Frontbridge, MX Logic). In the case of a customer using Securence's email filtering technology, SecurenceMail; whenever an email is sent to the customer's mail server the email is initially redirected to Securence through its MX record. MX record is short for ***m**ail **ex**change **record***, an entry in a domain name database that identifies the mail server responsible for handling emails for that domain name. The MX record points to an array of servers that runs in Securence's data center in Minneapolis and Milwaukee, figure 1. Before an email can be forwarded by Securence, a series of steps must occur to ensure "clean" delivery. This is known as filtering. According to Securence, 99% of all spam mails are detected by their filtering technology.

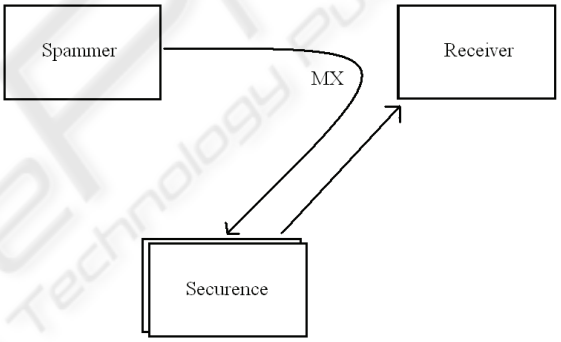

Figure 1: Filtering emails.

2.2 Filtering Emails from Hijacked Servers

In the case of hijacked computers, the owner of such systems, often organizations with high speed Internet connections and high processing power, may in fact protect themselves from being used for sending unsolicited emails. This can be achieved due to Frontbridge's technology. Frontbridge provides technology for this as shown in figure 2. Emails are filtered for spam and viruses at Frontbridge's Global Data Center Network (GDCN) before they are forwarded to the receiving mail server.

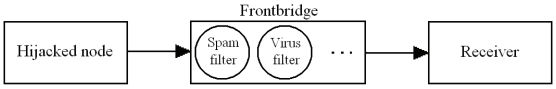

Figure 2: Frontbridge's filtering technology (simplified).

Utilizing both of these schemes for filtering emails, the receiver will not notice any additional processing load. However, the hijacked node in figure 2 is exposed of an additional load generating and/or sending these unsolicited emails. Such emails should be detected by Frontbridge, but still, the hijacked node suffers additional processing and transmission costs. As this is another problem, we will not discuss it any further in this paper.

2.3 Filtering Techniques

Often, filtering-processes filter spam, viruses, worms, junk mail, malicious content and attachments before reaching the end user. To detect spam, the message subject, sender, content of the email and attachments are checked for signs to determine whether the email actually is a spam mail.

Blacklisting of source IP addresses are in most cases checked once the email enters the filtering technology. Once the IP source has been authenticated, whitelist filtering is often applied. All emails from IP addresses on this whitelist are delivered directly to the recipient, bypassing spam filters. However, this only occurs if the source is not on the blacklist. Spam signature-tests are also often applied on the email to determine whether the incoming email possesses certain characteristics of a spam email. A rating is generated, and the email is market as a spam or not due to a threshold value. This approach may of course lead to false positives. Once an email has gone unmarked through the different spam filters, the email is often checked for viruses.

Virus filtering offered by Securence is deployed by technology provided by Norman AntiVirus (Norman) and Clam AntiVirus (Clam Antivirus). Norman not only disinfects an email, but also uses Sandbox technology to spot viruses that don't yet have a signature. Clam, on the other hand, which cannot disinfect, is useful in searching for viruses because of its open source architecture. Clam has advanced mechanisms that protect against new types of malware, including image and HTML exploits, as well as phishing attacks. By providing these two anti-virus technologies with a number of anti-spam filtering techniques Securence delivers a powerful email filtering solution.

Apart from the mentioned technologies in section 2.1 and section 2.2, there are several other proposals on how to prevent spam. Such proposals include use of tokens (Schlegel, 2005), challenge-response schemes (Spamarrest), pre-challenge schemes (Roman, 2005), graylisting (Harris, 2004), domain-based email authentication (Delany, 2005) and encapsulation of policy in email addresses (Ioannidis, 2003). Password-based systems (Cranor, 1998) and micropayment systems (Abadi, 2003) have also been proposed.

3 SECURING EMAILS

All in all there exist two major types of cryptography today, symmetric and asymmetric. We will in this section briefly describe both of these technologies applied on emails; we will also show their shortcomings when used for securing such applications.

3.1 Symmetric Cryptography

Starting in the 1970s, symmetric cryptosystems have been widely adopted both in military and academic communities as well as in the commercial market segment. An example of this is the Data Encryption Standard system (DES) which is still a vital component of many cryptographic protocols. DES and its descendant, Advanced Encryption Standard (AES) are examples of symmetric block ciphers which are used in symmetric cryptosystems. In such schemes, the two parties involved must share a secret key used for the encryption and decryption of emails. To manage this, the sender and receiver of the emails can meet in person to exchange a secret shared key.

Implementing such a cryptosystem for the Internet however, calls for a distribution scheme for distributing the symmetric session keys shared by the sender and receiver of the emails.

Such schemes have a major shortcoming when applied for securing email systems; it is not a very scalable solution when incorporated in an email application ranging outside a small group of users. Schemes that used the server approach for authenticating one of the parties to the other one quickly rendered the server overloaded as the amount of email users increased. Also, if the server is down, session-key distribution is impossible.

3.2 Asymmetric Cryptography

While a symmetric-key cryptosystem could have been used for an email system containing limited number of users, the 1990s Internet boom, and hence email use, would render it useless. Now schemes

that didn't require online servers for broking session keys to all users were suggested.

These schemes are often referred to as asymmetric or public-key infrastructure (PKI) systems. In a PKI system there exist a pair of keys for each user, one is private and the other one is public. The public key is signed by a Certificate Authority (CA) and kept in a verifiable certificate. The certificate may be kept at the node which the public key belongs to or in a directory server. After validating the certificate against a revocation list and validating the signature of the CA on the certificate, the email-sending node extracts the public key belonging to the receiving node. The email is now encrypted with this key and sent to the destination node. On reception, the receiver decrypts the email with its private key.

Using a PKI system to encrypt emails, it is believed that the spam problem would be history. This is partly due to the difficulties of locating certificates and hence public keys. It is believed that the net gain for a spammer would be less than the effort needed to manage sending the unsolicited emails. In reality however, it is believed that a global PKI would collapse under the administrative weight of certificates, revocation lists, and cross-certification problems. Certificates are not easily located due to the lack of standard directories that publishes these certificates. The CA must also be online, and the client must validate the received certificate, and match the certificate policy with the client's own policy requirements. This can be very time consuming. The size of the revocation lists may also become a problem as the client must check against them for deprecated certificates. Additionally, a global PKI would render political challenges. Would all the countries in the Middle East trust a root PKI CA located in the USA and vice versa? The PKI operations are shown schematically in figure 3.

Instead of requesting the certificate directly from Bob, the spammer could also request it from a directory server if available.

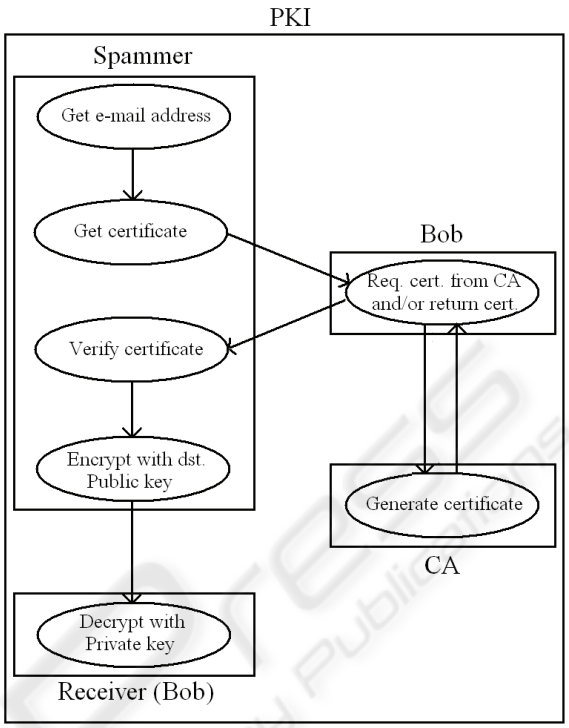

Figure 3: PKI operations.

As we see, there are lots of challenges both in the symmetric as well as in the asymmetric cryptography approach. Hence, separately, neither of these solutions gives life to a perfectly working and secure email scheme. New approaches are needed to solve the confidentiality and integrity problems of current email applications.

4 IDENTITY-BASED ENCRYPTION

In this section, we briefly review the identity-based encryption (IBE) and the Boneh-Franklin IBE scheme. Later on we will describe its use in securing email solutions.

4.1 Basic of IBE

The concept of identity-based cryptography was first proposed in 1984 by Adi Shamir (Shamir, 1985). In his paper, Shamir presented a new model of asymmetric cryptography in which the public key of any user is a characteristic that uniquely identifies the user's identity, like an email address. In such a scheme there are four algorithms: (1) setup generates global system parameters and a master-

key, (2) extract uses the master-key to generate the private key corresponding to an arbitrary public key string ID$\in\{0, 1\}^*$ (3) encrypt encrypts messages using the public key ID, and (4) decrypt decrypts messages using the corresponding private key.

The distinguishing characteristic of identity-based encryption is the ability to use any string as a public key. The functions that compose a generic IBE are thus specified as follows.

Setup: takes security parameter t_s and returns t_g (system parameters) and *master-key*. The system parameters include a description of a finite message space M, and a description of a finite ciphertext space C. Intuitively, the system parameters will be publicly known, while the *master-key* will be known only to the Private Key Generator (PKG).

Extract: takes as input t_g, *master-key*, and an arbitrary $ID\in\{0, 1\}^*$, and returns a private key K. Here ID is an arbitrary string that will be used as a public key, and K is the corresponding private decryption key. The Extract algorithm extracts a private key from the given public key.

Encrypt: takes as input t_g, ID, and $m\in M$. It returns a ciphertext $c\in C$.

Decrypt: takes as input t_g, $c\in C$, and a private key K. It return $m\in M$. These algorithms must satisfy the standard consistency constraint, namely when K is the private key generated by algorithm Extract when it is given ID as the public key, then $\forall\ m\in M$: Decrypt(t_g, c, K) = m where c = Encrypt(t_g, ID, c)

4.2 The Boneh-Franklin IBE Scheme

The scheme is based on IBE technique and proposed by Boneh and Franklin (Boneh, 2001). From here on we use Zq to denote the group $\{0, \ldots, q-1\}$ under addition modulo q. For a group G of prime order we use G^* to denote the set $G^* = G\backslash O$ where O is the identity element in the group G. We use Z^+ to denote the set of positive integers.

Algorithm 2.1 The full Boneh-Franklin IBE scheme:

1) Setup: Given a security parameter $k\in Z^+$, the algorithm works as follows.

Step 1: Run $\mathcal{G}$ on input k to generate a prime q, two groups G_1, G_2 of order q, and an admissible bilinear map $\hat{e}$: $G_1\times G_1\rightarrow G_2$. Choose a random $\alpha\in G_1$.

Step 2: Pick a random $s\in Z_q^*$ and set $\beta=\alpha^s$.

Step 3: Choose cryptographic hash functions for some n, H_1: $\{0, 1\}^*\rightarrow G_1^*$, H_2: $G_2\rightarrow\{0, 1\}^n$, H_3: $\{0, 1\}^n\times\{0, 1\}^n\rightarrow Z_q^*$, H_4: $\{0, 1\}^n\rightarrow\{0, 1\}^n$.

The message space is M= $\{0, 1\}^n$. The ciphertext space is C = $G_1^*\times\{0, 1\}^n\times\{0, 1\}^n$. The output system parameters are π = $\{q, G_1, G_2, \hat{e}, n, \alpha, \beta, H_1, H_2, H_3, H_4\}$. The master key is $s\in Z_q^*$.

2) Extract: For a given string $Id\in\{0, 1\}^*$ the algorithm
does:

Step 4: Computes $Q_{Id} = H_1(Id) \in G_1^*$.

Step 5: Sets the private key K_{Id} to be $K_{Id} = (Q_{Id})^s$ where s is the master key.

3) Encrypt: To encrypt $m\in$M under the public key Id do the following:

Step 6: Compute $Q_{Id} = H_1(Id) \in G_1^*$.

Step 7: Choose a random $\sigma\in\{0, 1\}^n$.

Step 8: Set $r=H_3(\sigma, m)$.

Step 9: Set the ciphertext to be

$$c = \langle r\alpha, \sigma\oplus H_2(g_{Id}^r),\ \mathrm{m}\oplus H_4(\sigma)\rangle,$$

where $\mathrm{g_{Id}} = \hat{e}(Q_{Id}, \beta)\in G_2$

4) Decrypt: Let c = <U, V, W> be a ciphertext encrypted using the public key Id. If $U\notin G_1^*$, reject the ciphertext. To decrypt c using the private key $K_{Id}\in G_1^*$ do:

Step 10: Compute $V\oplus H_2(\hat{e}(K_{Id}, U)) =\sigma$.

Step 11: Compute $W\oplus H_4(\sigma) = m$.

Step 12: Set $r = H_3(\sigma, m)$. Test that $U = r\alpha$. If not, reject the ciphertext.

Step 13: Output m as the decryption of c.

This completes the description of a full version of Boneh-Franklin IBE algorithm. More details about the security of the Boneh-Franklin IBE algorithm can be found in (Boneh, 2001, Boyen, 2003).

5 IDENTITY-BASED ENCRYPTION ON EMAILS

Shamir's original motivation for identity-based encryption (Shamir, 1985) was to simplify certificate management in email systems. When Alice sends an email to Bob at bob@company.com she simply encrypts her message using the public key string "bob@company.com". There is no need for Alice to

obtain Bob's public-key certificate. When Bob receives the encrypted mail he contacts a third party, which we call the Private Key Generator (PKG). Bob authenticates himself to the PKG in the same way he would authenticate himself to a Certificate Authority (CA) and obtains his private key from the PKG. Bob can then read his email.

Based on the new public-key cryptography using a commonly known identifier as the user's public key, Voltage has implemented a software agent called SecureMail. By utilizing the Boneh and Franklin IBE scheme, SecureMail can be used with existing email solutions and enable users to transparently send and receive their emails securely. The system eliminates the need for individual per-user certificates and the requirement to connect to a third-party server to verify these certificates before initiating secure emailing. Their solution is considered highly scalable because it eliminates the need for an additional infrastructure. Third-party CAs are not required and no information needs to be pre-shared. Using IBE for securing emails, the mails may even be encrypted or decrypted offline. Voltage's solution for email applications using IBE gives users the opportunities to conduct business securely from anywhere in the world. Voltage's SecureMail makes ad hoc business communication as easy as traditional non-encrypted emailing.

Though this seams very promising, our general concern regarding the use of IBE for securing emails is that a spammer more easily can manage to get hold of valid public keys. These keys can then be used for end-to-end email content encryption. Now, only the node associated with the public key, i.e. the email address, is able to decrypt the email. Hence no filtering services may be utilized before the mail is received at the destination node. This might be dangerous due to viruses and most bothering and time consuming due to spam. Of course such a scheme may also be used as a Denial of Service attack (DoS) as the recipient uses resources decrypting the received emails. Received emails should then subsequently be filtered for spam and viruses by the receiving node.

Note that unlike the existing secure email infrastructure, Alice can send encrypted emails to Bob even if Bob has not yet set up his public-key certificate; hence, it is even easier for Alice to get hold of potential victims. Also note that key escrow is inherent in identity-based email systems; the PKG knows Bob's private key. This might also compromise the security.

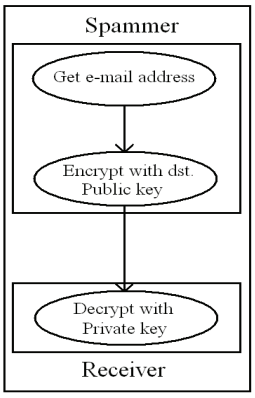

Figure 4: IBE operations (from a spammer's point of view).

As mentioned in the abstract, a large-scale use of IBE for securing emails on the Internet substantially increase the amount of spam possible for a spammer to send out during a given period of time compared to if a PKI solution is chosen. As shown in figure 4, the work required by a spammer is far less compared to if an ordinary PKI scheme as shown in figure 3 is chosen. The spammer may simply use automated processes to generate random public keys once in possession of an organization's email format. To succeed, the only thing the spammer must do is to encrypt the unsolicited emails using the victims email addresses.

5.1 Hijacking

As mentioned in section 2, a spammer may take advantage of different methods for sending unsolicited emails, e.g. by sending them itself or by capturing another node (hijacking). Also, as mentioned in that section, hijacking is an ever increasing way of doing spamming. By editing Frontbridge's scheme in a minor way as shown in figure 5, the *hijacked server spamming* can easily be stopped; now both for encrypted as well as unencrypted emails.

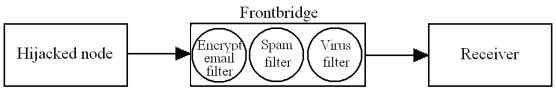

Figure 5: Filtering IBE encrypted spam.

For unencrypted unsolicited emails, the standard Frontbridge solution manages the filtering very well. For IBE encrypted ones however, the emails received at Frontbridge's network fails to be checked for spam and viruses due to the already existing IBE encryption. Therefore we recommend

such emails to be dropped and hence not forwarded by Frontbridge to the intended recipient.

This solution should be easy to integrate in Frontbridge's scheme. Hence, the hijacking way of sending unsolicited emails should be problematic for the spammer, that is, whenever the hijacked node is secured by technology provided by Frontbridge. As an organization now can protect itself from being used by a spammer for sending unsolicited encrypted emails, the spammer now has to do all the work by itself.

The Frontbridge solution, even though not fully described in this paper, uses symmetric encryption to secure the data sent from the email-sending node to the Frontbridge Global Data Center Network (GDCN). This is shown in figure 6.

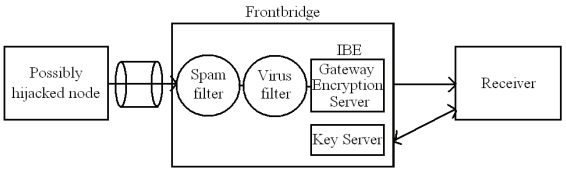

Figure 6: Frontbridge's current GDCN solution.

Figure 6 also shows that Frontbridge already provides IBE encryption from the GDCN to the receiving node.

5.2 The Problems of Securing Emails using IBE

There are as we see it mainly two disadvantages with the IBE solution when it comes to spamming. The first disadvantage is that a spammer may easily discover valid public keys (email addresses), or even use automated processes generating lots of public keys based on the knowledge of an organization's email format. Though easy discovery of public keys is the main selling point of IBE schemes applied on emailing, we see it from the spammer's perspective as a huge opportunity for pumping out vast volumes of unsolicited emails. Thereby the spammer manages to send more spam compared to if a PKI-like solution is chosen. This is partly due to the difficulties of locating certificates containing valid public keys.

The other disadvantage, which is an even greater concern, is the processing now required at the spam-receiving node. Once an IBE encrypted email is received, decryption is needed before the content may be checked for spam and viruses. Compared to currently used email solutions where usually no encryption is applied and filtering often is done by a third party, this may be used to launch a DoS attack on the receiving node. Compared to a PKI solution, the work required by the spammer is far less. The work required by the destination node is comparable to what is required in a PKI solution. As often when new technology occurs, the focus is on the benefits compared to existing technologies. However, IBE as we see it is more vulnerable to DoS attacks, spamming and the associated virus diffusion.

5.3 Spam Directly from the Spamming Node

As we have analysed the server-hijacking phenomenon of doing spamming, both for encrypted as well as unencrypted emails, we are left with the unresolved problem of IBE spamming directly from the spamming node. Currently this is an open problem. It would not be easy for the attacked node to know whether the incoming email is sent directly from a node (e.g. a spammer) or through a filter provided by a third party filtering out spam and viruses. So far, we have not succeeded in discovering a proper countermeasure to this form of spamming attacks.

One countermeasure would of course be to make the receiving node decrypt all incoming emails. This could be done in a sandbox to avoid possible infections from viruses. Decrypted emails could then be redirected to a service provider as Securence which supplies filtering technology as described in section 2. However, if the content of an email is to be secure, the decrypted email has to be encrypted once again, e.g. by a key shared by the destination node and Securence. The email may then be transmitted to Securence for filtering. Securence has to decrypt the email, filter it, and then encrypt it with the shared key. The email can then be sent back to the destination node which once again has to decrypt the incoming email. Now the email content can be read by the receiving node without the danger of being spammed or attacked by viruses from the originator of the email.

Though this is a possible solution, the required processing in this solution is immense, and hence not a good solution to the problem. Seen from a spammer's point of view, DoS attacks are in this scheme highly encouraged.

6 CONCLUSION

As often when new technology occurs, the focus is on the functionality of the technology and not on its security. In this paper we study some of the currently available technologies that provide spam and virus filtering on emails. We also study the effect of applying IBE on emails and the associated ease for a spammer to increase the amount of spam sent on the Internet.

Essentially, we have two main concerns about the use of IBE applied on emails. First, a spammer more easily manages to get hold of valid public keys to destination nodes. Second, denial of service attacks may be launched more successfully at a victim node due to the processing required to decrypt incoming emails. Filtering of spam and viruses also has to be done locally by the email-receiving node.

REFERENCES

Shamir, A., 1985. "Identity-based cryptography and signature schemes", *Advances in Cryptology*, CRYPTO'84, *Lecture Notes in Computer Science*, vol. 196, pp. 47-53.

Feige, U., Fiat, A., Shamir, A., 1988. "Zero-knowledge proofs of identity", *J. Cryptology*, vol. 1, pp. 77-94.

Fiat, A., Shamir, A., 1986 "How to prove yourself: practical solutions to identification and signature problems", In *Proceedings of CRYPTO'86*, pp. 186-194.

Boneh, D., Franklin, M., 2001. "Identity-based encryption from the Weil pairing", in *Advances in Cryptology*, CRYPTO 2001, *Lecture Notes in Computer Science*, vol. 2139, pp. 213-229.

Boyen, X., 2003. "Multipurpose Identity-based signcryption, a Swiss army knife for identity-based cryptography", in *Proceedings of the 23rd Interna. Conf. On Advances in Cryptology*, *Lecture Notes in Computer Science, vol.* 2729, pp. 383-399.

Chen, L., Kudla, C., 2002. "Identity-based authenticated key agreement protocols from pairings", *Cryptology ePrint Archive*, Report 2002/184, http://eprint.iacr.org/2002/184.

Lynn, B., 2002. "Authenticated identity-based encryption", *Cryptology ePrint Archive*, Report 2002/072, http://eprint.iacr.org/2002/072.

Waters, B., 2004. "Efficient Identity-Based Encryption Without Random Oracles", *Cryptology ePrint Archive*, Report 2004/180, http://eprint.iacr.org/2004/180.

Voltage security, 2004. E-mail Security – The IBE Advantage.

Veigner, C., Rong, C., 2006. "Identity-Based Key Agreement and Encryption for Wireless Sensor Networks", in preprint.

Veigner, C., Rong, C., 2006. "Simulating Identity-Based Key Agreement For Wireless Sensor Networks", in preprint.

DES (Data Encryption Standard), FIPS 46-2, http://www.itl.nist.gov/fipspubs/fip46-2.htm

AES (Advanced Encryption Standard), FIPS 197, http://csrc.nist.gov/CryptoToolkit/aes/

Securence, www.securence.com

Frontbridge, www.forntbridge.com

MX Logic, www.mxlogic.com

Norman, www.norman.com

Clam Antivirus, www.clamav.net, www.clamwin.com

Schlegel, R., Vaudenay, S., Dec. 2005. "Enforcing Email Addresses Privacy Using Tokens", In *Information Security and Cryptology LNCS 3822, First SKLOIS Conference (CISC 2005)*, pp. 91-100, Springer-Verlag.

SpamArrest, www.spamarrest.com

Roman, R., Zhou, J., Lopez, J., May 2005. "Protection against Spam using Pre-Challenges", In Security and Privacy in the Age of Ubiquitous Computing IFIP TC11, 20th International Information Security Conference (Sec'05), pp. 281-294, Springer-Verlag.

Harris, E., 2003. The Next Step in the Spam Control War: Graylisting. www.graylisting.org/articles/whitepaper.shtml

Delany, M., 2005. Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys). IETF Draft.

Ioannidis, J., Febr. 2003. Fighting Spam by Encapsulating Policy in Email Addresses, *Symposium on Network and Distributed Systems Security (NDSS 2003)*.

Cranor, L., LaMacchia, B., Aug. 1998. "SPAM!", *Communications of the ACM*, 41(8) pp. 74-83.

Abadi, M., Birrell, A., Burrows, M., Dabek, F., Wobber, T., Dec. 2003 "Bankable Postage for Network Services", *8th Asian Computing Science Conference*.