



Usable secure email communications: criteria and evaluation of existing approaches

Cristian Thiago Moecke and Melanie Volkamer
Fachbereich Informatik, T.U. Darmstadt, Darmstadt, Germany

Secure email communications

41

Abstract

Purpose – Email communication has been used for many years, and has begun to replace traditional, physical correspondence more and more. Compared to a traditional postal service, email services are easier, faster, and free of charge. Standard email, however, is, from a security point of view, more comparable to post cards than letters. Some security techniques and services exist, but few people use them due to lack of awareness, low usability, and a lack of understanding of Public Key Infrastructures (PKIs). A comprehensive comparison is missing, which makes it difficult for users to decide which email service to use. The purpose of this paper is to identify evaluation criteria covering security, usability, and interoperability aspects of email, and to apply them to existing email services.

Design/methodology/approach – The authors first define criteria based on literature review, threat analysis and expert consultation. These criteria are then applied, when applicable, to existing approaches including DKIM, SPF, PGP, S/MIME and Opportunistic Encryption, and to common secure email providers including Gmail, Hushmail, and De-Mail.

Findings – None of the existing analysed services meets all the derived criteria. Based on the result of the application of these criteria and the corresponding comparison, the authors propose future directions for usable secure email communication.

Originality/value – The criteria proposed are original and allow an evaluation and a comparison of different email systems that not only considers security aspects, but also the relation and trade-offs between security, usability and interoperability. Moreover, the trust assumptions involved are also considered.

Keywords Electronic mail, Information management, Data security, Secure email, Usable security, Criteria

Paper type Conceptual paper

1. Introduction

Ray Tomlinson sent the first network email in 1971 using the ARPANET. With the conversion from ARPANET to the internet in the early 1980s, email communication became broadly available. Since then, email has become more and more popular. Nowadays, email accounts are free of charge and much faster than traditional letters sent via postal services. Many people have several email accounts. Email replaces traditional letters more and more in many areas including in the private, business, and governmental sector. Critical, sensitive, personal and business information are sent via email although it is well known that emails are less secure than traditional letters. They can easily be forged; and provide neither strong sender authenticity nor message confidentiality. Secure email communication is in general available, i.e. based



Information Management &
Computer Security
Vol. 21 No. 1, 2013
pp. 41-52

This work has been funded by Horst Görtz Stiftung and by the Center for Advanced Security Research Darmstadt.

© Emerald Group Publishing Limited
0968-5227
DOI 10.1108/09685221311314419

on PGP or S/MIME. These techniques are far from being broadly used. Studies like (Whitten and Tygar, 1999; Sheng *et al.*, 2006) show that these solutions are not usable. People in particular do not understand the concept of standard public key infrastructures (PKI).

Solutions for more usable secure email communication have recently been proposed to improve the situation. Examples are the E-PostBrief (www.epost.de), De-Mail (www.bsi.bund.de) and Hushmail (www.hushmail.com/). These approaches do not provide end-to-end secure email communication but users need to trust the providers. As in addition some of these solutions are closed systems and users need to pay for them; it is unclear whether they will be accepted like common emails. As an optimal solution for usable and end-to-end secure email communication is not yet available, users need to find an adequate trade-off for their needs. This paper defines the most relevant criteria that should be taken into account for such a decision. We apply these criteria to analyse existing security mechanisms in the context of email communication (namely DKIM, SPF, PGP, S/MIME and TLS) and also to analyse how popular email providers (namely Gmail, Hushmail, E-PostBrief and DE-Mail) application of the previous techniques conforms to these criteria.

The paper is structured as follows: in Section 2, we analyse related work. In Section 3 we define the security criteria, which are applied to available security techniques discussed in Section 4 and existing email providers discussed in Section 5. In Section 6 we present our conclusions and directions for future work.

2. Related work

There is little published literature that proposes criteria for classifying and analysing the security of an email system. We analyse this work and how it compares to our work: NIST (National Institute of Standards and Technology) published Guidelines on Electronic Mail Security (Butterfield *et al.*, 2007). The guide “is intended to assist organizations in installing, configuring, and maintaining secure mail servers and mail client” (Butterfield *et al.*, 2007). The guide is based on available technology and on assisting administrators and users to apply this technology in the most secure way. Our work proposes criteria independent from available email systems and technologies. Alperovitch *et al.* (2007) analyses some of the previous work on email reputation systems (systems that calculate a score for an email, usually for spam filtering) and provide a “taxonomy that examines the required properties of email reputation systems, identifies the range of approaches, and surveys previous work” (Alperovitch *et al.*, 2007). While their work classifies and organizes a broad range of reputation criteria, they do not analyse whether current systems ensure these criteria. Garfinkel *et al.* (2005) focuses on usability aspects of email security, and includes a survey on how users act regarding signed emails. There are also previous works on usable security analysis of secure email systems (Whitten and Tygar, 1999; Garfinkel and Miller, 2005; Sheng *et al.*, 2006), which provide important usability criteria, which we also consider in our work.

3. Email usable security criteria

This section defines the evaluation criteria used to later analyse existing email security techniques and email providers. They include security, interoperability, and usability requirements. Those criteria are derived from a literature review, threat analysis and

expert consultation. Due to space constraints, we leave aspects like malware spreading, spam, traceability, long-term authenticity and confidentiality, legal aspects and anonymous communication including aliases for future work.

3.1 Security

We deduced the security criteria from a threat analysis of different email communication scenarios (i.e. business to client communication and communication between friends). These criteria are divided into sender authenticity (A), integrity (I), confidentiality (C), and one criteria addressing trust in the service under evaluation (Tr). We distinguish the following criteria:

- (A.1) Authenticity of sender address – it should be possible to verify that the sender is the owner of the email account that belongs to the stated sender email address – that is, that the sender really has access to the email account behind the declared email address.
- (A.2) Authenticity strength – it should be possible to verify if strong authentication methods were used, i.e. password, two-factor authentication, or asymmetric cryptography (this authenticity information, especially how strong the authentication method was, are of interest to the receiver of a message. However, none of the later evaluated approaches provide this kind of information to the receiver, therefore we analyse only if there are strong authentication mechanisms available or enforced).
- (A.3) Persona authenticity – it should be possible to securely identify the real identity of the person sending the email (from the email address or by other means, like digital signatures).
- (I) Integrity – it should be possible to detect modification of the content/the subject of the received email.
- (C) Confidentiality – it should not be possible to anyone else except the receiver of the email to gain access to the content/subject of the email.

All the previous criteria usually rely on one or more trust assumptions. Therefore, when analysing the fulfilment of the requirements, the necessary (Tr) trust assumptions should also be indicated.

3.2 Interoperability

Interoperability is important, as services are not ready for large-scale usage without it. Correspondingly, the sub criteria are related to security and trust aspects that are important for the acceptance of the corresponding service. These criteria were defined based on expert consultation (explanations are provided for each criteria). This class of criteria is divided into the following sub criteria:

- (IN.1) Provider interoperability – it should be possible to communicate with people who do not use the same email provider. This is required, as people do not tend to migrate and start using new providers when they cannot communicate with other people that may have decided to use other provider.
- (IN.2) Provider secured interoperability – it should be possible to securely communicate with people who do not use the same email provider.

The justification is the same as above, but considering that the user also has interest on communicating securely with people that choose other provider.

- (IN.3) Open standards – it should be possible to setup and run an own email service, without relying on existing providers. This is especially important to corporations, which often prefer to operate and have full control of their email services.
- (IN.4) Provider independency – there should be more than one provider offering the corresponding service. When the previous requirement is not fulfilled, having alternatives to choose facilitates adoption.
- (IN.5) Security extension – it should be possible to use the system with other security techniques discussed in Section 4 that are not implemented by default on the system (e.g. using add-ons). This also increases the possibility of acceptance of the system, since users may migrate and still communicate securely with people that did not migrate to the new or different technology.

3.3 Usability aspects

The usability criteria are deduced from literature review on usable security analysis of email communication systems (Whitten and Tygar, 1999; Garfinkel and Miller, 2005; Sheng *et al.*, 2006, 2010). The most important are:

- (U.1) Start-up cost – it should be easy to setup and start using the service.
- (U.2) Adequate metaphors – security functions like encryption, signing, and certificates should use adequate metaphors.
- (U.3) Easy trust decisions – users should only be confronted with few and easy to make trust decisions.
- (U.4) PKI is hidden – users should not need to understand the underlying PKI concepts (to make any trust decisions).
- (U.5) No out-of-bounds verifications – users should not need to obtain and/or compare some information via out of bounds mode like comparing hashes.

4. Security techniques available for email

In this section we describe the security techniques that are currently available and could be used to improve email security by integrating them in existing email services or by installing a corresponding plugin into the email client or web browser. The list contains techniques related to sender authentication, public key infrastructure (PKI) support, and securing the network. We analyse which of the above security and usability criteria they fulfil while we only discuss those security requirements that are at least partially ensured and only those usability requirements that can be applied but are not fulfilled. We also do not address interoperability requirements as the analysed techniques do not stand on their own but can only be used as an add-on to the email services discussed in Section 5. For all techniques we make the trust assumption (Tr) that the software running the corresponding security operations and checks is trustworthy, and that cryptographic secret keys are securely stored.

A sender policy framework (SPF) (RFC4408) record is a list of IP addresses that are authorized to send emails for a particular domain. The DNS server responsible for this domain publishes this list. This list can be used to partially verify the authenticity of a

received email, confirming that it came from an authorized server. If we trust the DNS server regarding the integrity of this list and the email provider behind this domain (Tr), i.e. the provider only delivers emails from authenticated users with the correct address in the “from” field, and there is also no MITM attack (a man-in-the-middle attack could modify the list since it is not authenticated) when fetching the list, A.1 is ensured. However, since this assumption is not true for all DNS servers and all mail providers, hard trust decisions remain for the user (U.3). *Sender ID framework – SIDF* (RFC4406) extends the SPF verification, but with no relevant improvements considering our criteria.

Domain keys identified mail (DKIM) (RFC4871) is a digital signature from the email provider on sent emails. It allows verifying that this provider really sent the message. The corresponding public key is published on the DNS server associated to this domain. Thus, regarding A.1, one needs to trust that the DNS server provides the proper key and that the email service only delivers emails from authenticated users. Correspondingly, the trust assumptions (Tr) regarding A.1 and the limitations for U.3 identified for SPF also hold for DKIM, including the non-occurrence of a MITM attack (in this case the MITM attack could be used to inform the wrong key to the receiver). In addition, the integrity property (I) is partially ensured because the signature provides message integrity after being processed by the email server of the sender under the same trust assumption required for A.1.

Pretty good privacy (PGP) (RFC2015) provides end-to-end security to email messages. It is not necessary to trust a central trust anchor as it is based on the web-of-trust (WoT) model. It fulfils property C and I. Authenticity depends on the adequate use of the WoT. In a restricted environment where people verify carefully the keys they sign, A.1 and even A.2 can be considered as fulfilled, but with the cost of hard trust decisions (conflict with U.3) and the fact that out-of-bounds data (i.e. verifying key hashes before setting keys as trusted) is required (conflict with U.5). PGP requires some PKI understanding (conflict with U.4) and it takes a while to get started (conflict with U.1). Adequate metaphors (U.2) are very important for a user friendly implementation of PGP, while studies (Whitten and Tygar, 1999; Garfinkel *et al.*, 2005) showed that this is not the case for PGP (conflict with U.2).

Secure/multipurpose internet mail extensions (S/MIME) (RFC2311) provides end-to-end security to email messages. It is based on X.509 PKIs. A certificate for a corresponding key pair should be obtained from a trusted Certification Authority (CA) which is usually not free of charge. However, the use of self-signed certificates is also possible. S/MIME with CAs fulfils A.1, C and I, and if the CA also verifies the personal identity (i.e. for Qualified Certificates on EU (RFC3739)) it fulfils A.2, as well. The receiver needs to trust the CA that issued the certificate of the sender (Tr). While major CAs are already set as trusted on most plugins, it leads to hard trust decisions if the sender’s certificate is issued on an untrusted/unknown CA (conflict with U.3). Without CAs (self-signed certificates), S/MIME only fulfils C and I. In this case it is hard to decide if the authenticity of the message should or should not be trusted (conflict with U.3). The user needs out-of-bounds information to verify the authenticity (conflict with U.5). In both cases, it takes a while to get started (conflict with U.1). Similar to PGP, adequate metaphors are very important for a user-friendly implementation. In addition, in cases in which the verification fails (i.e. failure on obtaining revocation information, or an intermediate CA certificate), it is required to understand the underlying PKI (conflict with U.4).

Opportunistic encryption (RFC4322) in general means that a system tries to use encryption to secure any communication, while using unencrypted communication if corresponding keys are not available. The keys are not necessarily authenticated. Thus, if there is any known key to communicate with the receiver (even in a self-signed certificate) encryption is used. Garfinkel (2003) proposed a system that implements opportunistic encryption for email communication. It acts like a proxy for sending and a proxy for receiving an email (while he does not provide clear information about the proxy, we assume that it runs as a plugin locally and is considered trusted (Tr)). The proxy manages the keys (including certificates), the encryption and the decryption process. Whenever the sending proxy is able to encrypt an email to the receiver, it will encrypt this message. It is also possible to specify that a message should only be delivered if it can be encrypted. There is no authentication of the keys in the system. The receiver proxy learns new keys from emails previously received and stores them in its own database. It gives a warning only if a new key is detected for an already-known email address. Furthermore, emails are not signed. It therefore provides confidentiality (C) under the trust assumption (Tr) that the proxy is trustworthy (i.e. a local proxy) and there was a previous legitimate communication between sender and receiver so the correct key is known. This idea avoids the need to understand PKI concepts (U.4 is ensured) and hard trust decisions (U.3 and U.5 is in general ensured – except when a conflict occurs). There is also a low setup effort (U.1).

Transport layer security (TLS) provides a secure communication between user and server (over IMAP, POP, SMTP or a Webmail session) and between servers. Confidentiality (C) and integrity (I) are ensured under the trust assumption (Tr) that the whole delivery process is secured by TLS and that the servers are trustworthy. It could also be used to authenticate the user with TLS client authentication (therefore offering strong authentication – A.2), but that is not common.

5. Analysis of existing secure email providers

In this section, we analyse five types of email providers according to our criteria while focusing on one representative of each; namely standard (insecure) email providers, very popular email providers providing some security (Gmail), a combination of the previous and the different PKI approaches from Section 4, providers offering secure email communication in a closed system (DE-Mail/E-Postbrief), and providers offering secure email communication in an open system (Hushmail). The results of the comparison are shown in Figures 1-3.

5.1 Standard email

According to the email standards (RFC5322, RFC5321), standard email services are not required to provide too much security information to the receiver. The only somehow relevant security information is the header of an email which provides information regarding the delivery path of this message, but this information is not authenticated, therefore any decision based on it depends on assuming all the servers on the path trustworthy. However, these assumptions are hard to verify and rely on too many out-of-bounds information (conflict with U.5) and hard trust decisions (conflict with U.3), and also are not true for many servers. Many email providers operate using no more security than those offered by these standards.

| | A.1 | A.2 | A.3 | I | C |
|----------------|---|------------------------------------|------------------------------|---------------------------|---|
| Gmail | Partially (see U.3 Tr: Gmail/DNS). Detect forged Gmail(Tr: Gmail) | 2 factor (not visible to receiver) | No | Yes (Tr: Gmail and DNS) | End to Server and Server to Server (if available) |
| DE-Mail | Yes (Tr: Provider) | Unknown | Yes (Tr: Provider) | Yes (Tr: Provider) | Yes (Tr: Provider) |
| E-PostBrief | Yes (Tr: Provider) | SMS TAN (not visib. to receiver) | Yes (Tr: Provider) | Yes (Tr: Provider) | Yes (Tr: Provider) |
| Hushmail | Yes (Tr: Hushmail) | Password only | No | Yes (Tr: Hushmail/Softw.) | Yes (Tr: Hushmail/Softw.) |
| Gmail & PGP | Restricted contexts (no U.5) | No change | Restricted contexts (no U.5) | Yes | Yes |
| Gmail & S/MIME | If CA verifies it (Tr: CA) | No change | If CA verifies it (Tr: CA) | Yes | Yes |
| Gmail & Opp | If had previous secure comm. | No change | No | No | If receiver key is known |

Figure 1.
E-mail services and security criteria

| | U.1 | U.2 | U.3 | U.4 | U.5 |
|----------------|--|---------------------------|----------------------------------|--|--|
| Gmail | Easy | No metaphors | Decide if domain is trustworthy | No PKI | Domain trust worthiness |
| DE-Mail | Identification on provider | Unknown | Easy: Trust provider | PKI hidden | No out of bounds data |
| E-PostBrief | Identification on Post office | Encryption and Signature | Easy: Trust provider | PKI hidden | No out of bounds data |
| Hushmail | Easy | Encryption and Signature | Easy: Trust provider | PKI hidden | No out of bounds data |
| Gmail & PGP | Hard: create /publish key, get recipient key | Depends on implementation | Decide trusted keys based on WoT | Understand PGP | Obtain information to trust keys |
| Gmail & S/MIME | Hard: obtain own/recipient certificate | Depends on implementation | If known CA, easy. If not, hard | If verification fails or CA is untrusted | If verification fails or CA is untrusted |
| Gmail & Opp | Easy | Not necessary | Only in key conflict | PKI hidden | Only in key conflict |

Figure 2.
E-mail services and usability criteria

| | IN.1 | IN.2 | IN.3 | IN.4 | IN.5 |
|-----------|----------------|-----------------|-----------------|----------------|-----------------|
| Gmail | Yes | Yes | Open standards | Open standards | PGP, S/MIME |
| DE-Mail | Closed environ | Closed environ. | High cost | High cost | PGP/S/MIME |
| E-PostBr. | Closed environ | Closed environ. | Closed environ. | No | PGP/S/MIME |
| Hushmail | Yes | Yes | Open standards | Open standards | DKIM/SPF/S/MIME |

Figure 3.
E-mail services and interoperability criteria

5.2 Gmail

The most popular email providers usually offer at least some security improvements, even if mainly for spam control. We have chosen to analyse Gmail, which verifies SPF and DKIM for email origin authentication, and adds DKIM signatures for emails delivered. When an email has SPF or DKIM authentication but the domain in the address (“from” field) does not match the authenticated origin information, this is shown

to the end-user as a “via authenticated origin domain” after the sender’s address. Emails sent from forged Gmail accounts (i.e. from other servers, but with Gmail address on the “from” field), which as a result do not have authenticated information on their origin, are shown to the receiving Gmail user with a warning this message may not have been sent by: email@domain.com, in red. This makes impersonating other Gmail accounts difficult.

Emails from domains other than Google and without SPF and without DKIM authentication are shown without any hint or warning. The only difference between a non-authenticated and an authenticated email is that the authenticated domain is shown as mailed-by when the user looks at the details of the message header. There is no other visual clue that differentiates an SPF/DKIM authentic message from one without any authentication. Therefore, Gmail only guarantees email proof of possession (A.1) between Gmail users, considering Gmail trustworthy (Tr). When communicating with other providers, the interface does not show adequate feedback, but even with more information provided it would lead to complex trust decisions (conflicts with U.3), namely deciding if the authenticated domain is trustworthy or not. Gmail supports two-factor authentication (using a smartphone application that generates “one time passwords”) but this is not enforced (A.2) and not visible to the receiver. Emails originated from Gmail servers are DKIM signed, therefore have their integrity guaranteed (I), considering Gmail trustworthy (Tr). Gmail uses SSL/TLS for end-to-server encryption, and may use it also to communicate with other servers when supported, however even trusting Gmail it is not possible to guarantee confidentiality of messages after they leave Gmail servers. Therefore, (C) is only partially fulfilled. The communication between user and server is encrypted during the authentication process (A.2).

Gmail uses only open standards and is a standard email provider (IN.1). All the security measures available are also based on open standards and may be used by other providers, so it is possible to have the same level of secure email communication with any other interested provider (IN.2). While there is only one provider of the Gmail solution, we consider IN.4 and IN.5 fulfilled because there is no restriction to operate a similar service once it is based on open standards. It is also possible to integrate PGP and or S/MIME on Gmail (there are even plugins available for that, i.e. FireGPG and Penango), therefore IN.5 is also fulfilled.

5.3 DE-Mail/E-PostBrief

DE-Mail and E-PostBrief are closed messaging systems, provided by the German Government and German postal services. These approaches claim to provide secure email communication and are based on strictly controlled servers that manage the full delivery process. There is also a law regulating DE-Mail. Both systems are closed environments. Thus, there is no possibility to forge a message (A.1), if the providers are considered trustworthy (Tr). Users are only allowed to use the system after a personal identification (A.3) at the provider’s office (DE-Mail) or at a postal service centre (E-PostBrief). E-PostBrief provides SMS TAN (a one-time-password sent via SMS) as a second factor of authentication, but its use is neither enforced (A.2) nor available to the receiver. There is no information available whether DE-Mail providers will offer this possibility. The communication between different servers of one provider and servers from different providers (in particular within the DE-Mail concept) are

secured by TLS. TLS is also used to provide end-to-server security. Correspondingly, integrity (I) and confidentiality (C) are fulfilled based on trust in the provider (s) (Tr). Besides the costs for each email (both services require payment per message sent), interoperability is the greatest disadvantage of these two approaches. Due to their closed environment property, it is only possible to communicate with people who have an account in this environment (IN.1 and IN.2 are not ensured). While there are some providers in DE-Mail, the number is limited (IN.4 insured) though it is possible to operate your own provider (with high costs involved – IN.3 ensured). It could be possible to use other techniques (IN.5), particularly PGP or S/MIME to provide end-to-end security, using corresponding plugins.

5.4 Hushmail

Hushmail is a web-based solution (it is also possible to use a Java applet to download the encrypted key and process the encryption operations locally). However, the same trust assumptions hold for the Java version, since Hushmail provides it. Hushmail is also testing a new interface, but it is not stable. Therefore, our evaluation considers the old (“original”) interface. Hushmail calls itself a PGP-based secure email system. However, the main characteristic of PGP, namely the WoT model, is not applied. Instead, Hushmail signs all the public keys of the users and publishes them on a key server. Obviously, all keys signed by Hushmail are considered trustworthy. With this approach, it is possible to automate the key trust management process and in particular the user is no longer involved. A Hushmail server generates the user’s key pairs and the secret key is encrypted with the user email account password. In addition, everyone is able to publish new PGP keys on the key server, associated to email addresses from other email providers besides Hushmail. In that case, there is a challenge-response email sent to the email associated with the key to verify the ownership of the account before the key is set as trusted for all Hushmail users. It is also possible to send secure emails to non-Hushmail accounts for which no PGP key is stored on the Hushmail key server, based on question/answer that only the receiver should be able to answer.

For all received signed emails for which the corresponding public key is trusted by Hushmail, the signature is verified and if the email is properly signed the email is shown as having valid signatures (“This message is encrypted, and is digitally signed by Sender Name <username@domain.com > ”), which fulfils A.1. Emails are not verified using DKIM or SPF. There is only a warning if the “return-path” field does not match the sender address. Since this information may be forged, there are no other security indicators of authenticity for non-signed emails that are received. The user’s identity is not verified before creating an account (A.3 fails) and the user is authenticated by password.

Email integrity (I) and confidentiality (C) are in general ensured under the assumption that Hushmail is trustworthy (Tr) (due to space constraints we only consider the case where PGP is in place while there are some usability problems with the implementation of the question/answer based encryption). It is possible for the user to verify message integrity also using its own trusted software and verifying the keys using out-of-bounds data. Since it operates using open standards and can send standard insecure email, all interoperability criteria are fulfilled. The main usability problem is the lack of

adequate metaphors for cryptographic operations (U.2) and the communication with users without Hushmail account.

6. Discussion and conclusions

In this paper, we proposed criteria, which a usable and secure email system should ideally ensure. In particular, these criteria evaluate the security properties and how those relate with trust assumptions, interoperability and usability aspects. The list only contains the most important security and usability requirements due to space limits, but more should be considered on the development of an ideal solution.

We applied these criteria to email security techniques and also to existing email providers while discussing different groups of email providers including a combination with available security techniques as add-on. None of them ensures all the requirements.

Closed and web-based systems like Hushmail are more usable, but they do not provide end-to-end security, as one needs to trust the provider. On the other hand, the consequences of solutions based on security add-on for standard email services are hard trust decisions for the user. It also takes a while to set up these systems. In addition, trust in this security technique is also required, as the average user cannot develop this on his own. The opportunistic encryption approach is also more usable, however it does not offer authentication. Approaches similar to Hushmail also have the benefit that secure email can be used from any computer while the PGP/SMIME add-ons require you to carry the key in a secure way. The main disadvantage of DE-Mail/EPostbrief is that these are closed systems, i.e. it is not possible to communicate electronically with people who do not have such an account, and you need to pay for each email. While with Hushmail it is possible to communicate with people who do not have Hushmail both in a secure and insecure way, there are still some security and usability problems in the implementation.

For future work, we propose to merge these approaches. Combine systems like Hushmail with opportunistic encryption to hide the PKI and other security trust decisions even more from the user. Registration should be based on in person identification and authentication. Authentication should be based on two-factor authentication at the least. The type of authentication used by the sender should be provided to the receiver. While the key should be generated at the user's side, the email provider could store the encrypted key to make it accessible from any computer. Key management, distribution, and revocation could be based on S/MIME, but operated by the email servers rather than a third party CA. Communication to users without such an account could be based on an improved implementation of the Hushmail approach.

The biggest challenge seems to be on usability aspects, especially on trust decisions that the receiver needs to take. Future work should consider this carefully and reduce the necessary trust decisions. The software on the client side must be prepared to offer clear information about the security and trustworthiness of an email, so the user can decide how to react to the content of the message. The same applies for sending a message, the sender should be provided with enough information about how secure the delivery process will be.

The proposed criteria focus on aspects that may facilitate the decision of the receiver about the trustworthiness of an email. If we are able to securely authenticate

the origin of an email, we can also use this information for spam filtering and also to display more precise and contextualized warnings, e.g. regarding phishing emails and dangerous attachments.

References

- Alperovitch, D., Judge, P. and Krasser, S. (2007), "Taxonomy of email reputation systems", *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, IEEE Computer Society, Washington, DC, p. 27.
- Butterfield, J., Tracy, M. and Jansen, W. (2007), *Guidelines on Electronic Mail Security, Recommendations of the National Institute of Standards and Technology*, National Institute of Standards and Technology, Gaithersburg, MD.
- Garfinkel, S. (2003), "Enabling email confidentiality through the use of opportunistic encryption", *Proceedings of the 2003 Annual National Conference on Digital Government Research*, pp. 1-4.
- Garfinkel, S. and Miller, R. (2005), "Johnny 2: a user test of key continuity management with S/MIME and Outlook Express", *Proceedings of the 2005 Symposium on Usable Privacy and Security*, ACM Press, New York, NY, pp. 13-24.
- Garfinkel, S., Margrave, D., Schiller, J., Nordlander, E. and Miller, R. (2005), "How to make secure email easier to use", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, pp. 701-10.
- Sheng, S., Broderick, L., Hyland, J. and Koranda, C. (2006), "Why Johnny still can't encrypt: evaluating the usability of email encryption software", *Symposium on Usable Privacy and Security*, pp. 3-4.
- Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F. and Downs, J. (2010), "Who falls for phish?", *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI'10*, ACM Press, New York, NY, p. 373.
- Whitten, A. and Tygar, J. (1999), "Why Johnny can't encrypt", *USENIX Security*, Vol. 1999, p. 1.

Further reading

- Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J. and Thomas, M. (2007), DomainKeys Identified Mail (DKIM) Signatures. RFC 4871 (Proposed Standard).
- Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L. and Repka, L. (1998), S/MIME Version 2 Message Specification, RFC 2311 (Historic).
- Elkins, M. (1996), MIME Security with Pretty Good Privacy (PGP). RFC 2015 (Proposed Standard).
- Klensin, J. (2008), Simple Mail Transfer Protocol. RFC 5321 (Draft Standard).
- Lyon, J. and Wong, M. (2006), Sender ID: Authenticating E-Mail, RFC 4406 (Experimental).
- Resnick, P. (2008), Internet Message Format, RFC 5322 (Draft Standard).
- Richardson, M. and Redelmeier, D. (2005), Opportunistic Encryption Using the Internet Key Exchange (IKE). RFC 4322 (Informational).
- Santesson, S., Nystrom, M. and Polk, T. (2004), Internet X.509 Public Key Infrastructure: Qualified Certificates Profile. RFC 3739 (Proposed Standard).
- Wong, M. and Schlitt, W. (2006), Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental).

About the authors

Cristian Thiago Moecke has been a PhD researcher at Center for Advanced Security Research Darmstadt (CASED) Since August 2011. As a member of the group “SecUSo – IT Security, Usability and Society”, and under the supervision of Professor Dr Melanie Volkamer, the focus of his PhD work is usable secure email communication. He holds a Bachelor’s degree in Computer Science from the Federal University of Santa Catarina (Brazil), and a Masters in Information Security from the same university. Before starting his PhD at CASED he participated actively in the main academic and government efforts in the PKI area of Brazil. Cristian Thiago Moecke is the corresponding author and can be contacted at: cristian.moecke@cased.de

Prof. Dr Melanie Volkamer has been an Assistant Professor at the Department of Computer Science of Technische Universität Darmstadt (Germany) since February 2012 and has headed the research group “SecUSo – IT Security, Usability and Society” since the end of 2010. She received her PhD from the University of Koblenz in October 2008.