Challenges with End-to-End Email Encryption

Jiangshan Yu, Vincent Cheval, and Mark Ryan

School of Computer Science,
University of Birmingham, UK

1 Introduction

In recent decades, numerous efforts have been put on improving the security of email system. However, in practice it is rather hard for users to secure their mails when they are facing various threats, e.g. attacks from 'trusted' service providers (such as email servers or certificate authorities), mass surveillance from government agents, and malware controlled devices. Traditional encrypted email systems (i.e. S/MIME and PGP) exist but have failed to take off [1], which is in marked contrast with the encryption for the web, where encrypted browsing is routinely done by billions of users each day.

With the disclosure of NSA global surveillance, the confidentiality of exchanged messages and email user's privacy gain more focus from academics, industrial researchers, and public. In the last year, some companies who tried to provide secure email services had to choose between shutting down their services and revealing users' plaintext emails. For example, Lavabit, Edward Snowden's email provider, has chosen to shut down its service to prevent being forced by the NSA to sabotage its own encryption [2–4]. Silent Circle, another secure communication service, also shut down its email product almost at the same time as Lavabit [5]. PrivateSky, a British secure email service, was shut down also in 2013 because of the pressure from GCHQ [6].

These encrypted email service providers were forced to provide customers' email in clear to the government agent because that they have a way to get user's plaintext email. For example, the main idea of Lavabit protocol is that the Lavabit email server creates public/private key pairs for its customers, encrypts the private key with the corresponding customer's login password, and stores it on the server. For each incoming email, the Lavabit server encrypts the email with the receiver's public key before storing them on the server. When a customer wants to retrieve her email, she sends her password to the server to decrypt her private key, then the server uses the private key to decrypt the email, and sends the plaintext to her. In other words, the Lavabit behaves as a trusted third party for its customers, and they can have a copy of user's email in clear if they want. So, to have a better security guarantee, secure email services are desired to provide end-to-end encryption while without requiring a trusted third party.

To have a clear view, we list following challenges related to end-to-end email encryption systems:

– **Usability**. E2e email encryption has been achieved in different ways. However, existing ways (S/MIME and PGP) are not user friendly [1, 7].

- **Metadata protection** is a requirement related to user privacy since metadata of emails could leak sensitive data to third parties.
- **Key loss mitigation** is a requirement that aims to reduce damage when a long term secret key is revealed.
- **Spam and phishing mail detection over encrypted email** is another challenge. The traditional server side spam filter does work since the server should not be able to get user email in clear text. On the other side, the client side spam filter would be a burden for users.
- **Secure webmail** is the requirement to support e2e email encryption in webmail. This could be done by installing plug-ins into web browsers. However, the challenge is how a user can verify whether the plug-in or dynamically shipped code is secure and authentic.

2 ConfiMail

ConfiMail [7] is a recent proposal to provide e2e email encryption, claimed to be user friendly, while without requiring a trusted party. The main idea of ConfiMail is that each service provider maintains its own public log, and records their customers' public key in it. A sender's email client automatically queries and accepts a receiver's public key if it is in the log. In this way, all issued public keys will be recorded in the log, so the misbehaviour of a service provider will be detected by users. Hence, the service provider cannot cheat without leaving evidence of its having cheated. Since service providers don't want to lose their reputation in order to keep their customers, it is reasonable to assume that they will not launch such detectable attacks.

The security of ConfiMail relies on improved certificate transparency. Certificate transparency [8] was proposed by Google to mitigate the problem of misissued certificates by recording all issued certificates in publicly auditable, append-only, untrusted logs. Users should only accept certificates which is included in the log. The public log in certificate transparency is organised as a Merkle tree [9], which stores all certificates in the leaves, and new certificates are appended by extending the tree to the right. So given the hash value of the current root, users can verify if a certificate is in the tree in logarithmic time. Similarly, a user can efficiently verify if the received log is an extension of a previous version whose hash value of the root has been stored by the user. So interested parties can detect misused certificates by checking the log. Hence, the use of public log exposes misissues rather than prevent them.

However, such public log structure cannot provide the proof that a certificate has been revoked. To offer efficient certificate revocation verification, ConfiMail improves the log structure by using an additional Merkle tree but organised as a binary search tree. In which certificates with the same subject are stored in every nodes (including leaves) such that a left-right traversal yields the data in lexicographic order of subject. In particular, each node is a set which contains the user's identity and a list of public keys. A single bit "0" or "1" is appended after each public key to show whether the public key is revoked, for example,

"0" for valid public key and "1" for revoked public key. Hence, users can easily verify the validity of a public key. However, one still has to verify that these two Merkle trees are presenting the same set of data. This verification requires $O(n)$ time and space, but it does not have to be computed by any particular user's browser. There are two ways that can be used to achieve this efficiently.

- *Random checking by users' client software.* The client software specifies a randomly chosen path of a leaf in the first Merkle tree, terminating in a hash value of a certificate. Then it requests proof that this certificate is also included in the second Merkle tree.
- *Public monitor.* The monitor receives all the updates from a service provider and maintains its own version of the two trees. It compares the local hash value of tree root with the one reported by the log. Anyone can be a public monitor.

Therefore, an e2e email encryption is achieved without requiring any trusted party. In addition, since all public key queries and verifications are automatically done by email clients and all encryption and decryption processes are done in the background, users can use ConfiMail service as what they do today. Thus it is claimed to be user friendly.

3 Challenges for secure email

3.1 Usability

End-to-end (e2e) email encryption has been offered by existing email systems in different ways. S/MIME and PGP are two well known systems to offer e2e email encryption. In S/MIME, a user's public key needs to be certified by a trusted third party, called certificate authority, which is assumed to be trusted. In real world, however, it is hard for users to verify if the certified public keys are authentic, since the fake certificates may be issued by malicious or compelled certificate authorities.

PGP does not require a trusted third party to certify public keys. Instead, PGP spreads the certifying role across a set of users, each of whom are somewhat trusted and somewhat known to the sender and receiver, with the expectation that, taken together, this comprises enough evidence for the authenticity of the public key. By signing each others keys in a peer-to-peer fashion, PGP users create a web of trust that works not because of some highly trusted pillars like CAs, but because all the users support the trust web in a small way.

In spite of support on all major client software and significant efforts at supporting take-up, very few people use encrypted mail. Yet, there are substantial motivations, including compliance requirements as well as confidentiality requirements. End-to-end encrypted mail seems to have a dedicated following among a small number of people in very specific sectors. "Why Johnny can't encrypt" is a 1999 classic paper [1] explaining why PGP encryption for email has failed to take off. Other papers have developed the explanation further. Mainly, the reason is that it is too complicated for users to understand the model and to manage the keys.

3.2 Metadata protection

The metadata of emails leaks information (e.g. the identity of the sender, the receiver, the mail sending time, etc.) which could be sensitive for users. For example, by analysing all email metadata of a company, the customer network of this company can be discovered. So the metadata is desired to be private against third parties. Tor, an online anonymity service, could be a way to protect metadata. However, since the SMTP server can learn the metadata, so users may need to set up their own server rather than relying on the existing cloud SMTP servers. This will be a difficulty for common users.

How to protect metadata is a very hard challenge and ConfiMail does not have a way to address it.

3.3 Key loss mitigation

With existing secure email systems (e.g. S/MIME, PGP, and ConfiMail), if a user's secret key is somehow corrupted (e.g. the user has used the secret key in a malware controlled device), then possibly all previous emails of this user are revealed. Moreover, a user has no way to know if his key is leaked. In this case, the user's email can be read by the attacker until the user's key pair is expired – which may take couple of years.

We seek a way to maximally protect user emails when the decryption key is somehow corrupted. We call such a requirement 'key loss mitigation' i.e. the leak of a long term secret key only harms a part of user's previous emails, but not all of them. A special case of this requirement is known as perfect forward secrecy, in which the leak of a long term secret key will not reveal any previous encrypted message. Backward secrecy is another special case of this requirement – the leak of a long term secret will not reveal future encrypted messages.

Such requirements are difficult to be achieved in email systems, because that unlike other message protocols (e.g. off the record) with similar requirements, the email systems is asynchronous and email users want to go back to read their past emails. In other words, in email systems, we have not only a requirement for the message transmission, but also a requirement for message storage. So the existing methods for perfect forward secrecy are not well suited into email systems.

3.4 Spam detection

Spam and philshing mail detection over encrypted mail is another challenge since the email server cannot understand the content of an email, thus the existing spam filters could not work in the traditional way. One possible idea is that users send a white list to their email servers to specify the address that they want to receive encrypted mail from. However, the main concern of this idea is that the white list would leak the customer's network.

3.5 Webmail security

Most webmail applications (such as Google mail, Hotmail, and Yahoo mail) do not natively support secure email service (e.g. S/MIME and PGP). To use secure email service through webmail, one way is to install a third party JavaScript. However, the challenge is how users can be confident that the installed JavaScript is secure.

4 Conclusion

ConfiMail contributes substantially to addressing usability challenge in e2e encrypted email systems. However, the solution for other challenges are left as future work. By participating in the workshop, we hope to get opinions on which of them are the most important, and what approaches are practical.

References

1. A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0," in *Proceedings of USENIX Security Symposium*, 1999.
2. "Encrypted email provider Lavabit shuts down, blames us gov't," Slashdot, 2013. [Online]. Available: http://yro.slashdot.org/story/13/08/08/1956215/encrypted-email-provider-lavabit-shuts-down-blames-us-govt
3. "Lavabit forced to shut down," Slashdot, 2013. [Online]. Available: http://yro.slashdot.org/submission/2864767/lavabit-forced-to-shut-down?sdsrc=rel
4. "Joining lavabit et al, groklaw shuts down because of nsa dragnet," Slashdot, 2013. [Online]. Available: http://yro.slashdot.org/story/13/08/20/0750237/joining-lavabit-et-al-groklaw-shuts-down-because-of-nsa-dragnet?sdsrc=rel
5. "Silent Circle follows Lavabit by closing encrypted e-mail service," Slashdot, 2013. [Online]. Available: http://it.slashdot.org/story/13/08/09/1228255/silent-circle-follows-lavabit-by-closing-encrypted-e-mail-service?sdsrc=rel
6. "UK secure email provider shut down his service in january to prevent GCHQ from obtaining encryption keys," Techdirt, 2013. [Online]. Available: http://www.techdirt.com/articles/20131212/10563525549/uk-secure-email-provider-shut-down-his-service-january-to-prevent-gchq-obtaining-encryption-keys.shtml
7. M. D. Ryan, "Enhanced certificate transparency and end-to-end encrypted mail," in *NDSS*, 2014.
8. B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," RFC 6962 (Experimental), Internet Engineering Task Force, 2013.
9. R. C. Merkle, "A digital signature based on a conventional encryption function," in *CRYPTO*, 1987, pp. 369–378.