Cryptography for Secure E-mail Communication

Vibha Ojha[1], Ravinder Singh[2]
Govt. Engineering College, Ajmer
[1]*vibha.ojha@gmail.com*, [2]*ravikaviya@gmail.com*

Abstract: Users share private information on the web through a variety of applications, suchas email, instant messaging, social media, and document sharing. Unfortunately, recentrevelations have shown that not only is users' data at risk from hackers and malicious insiders,but also from government surveillance. This state of affairs motivates the need for users tobe able to encrypt their online data specifically the e-mail communication. This paper shows the use of cryptographic algorithms for secure e-mail communication.

Keywords: *Cryptography, E-mail, Privacy, Confidentiality*

*****

1. Introduction

While not necessarily an every-dayoccurrence, from time to time individuals need to use email to send highly sensitiveinformation. For example, some businesses will ask that a job applicant send their socialsecurity number to the business in order to process a reimbursement.

Furthermore,unlike many other communication mediums, email is usually archived by the email

server, increasing the vulnerability of sensitive information sent over email.While there are some features that are unique to secure email,many of the principles for designing securing email in a usable manner also apply toother content-based encryption systems.

While new communicationplatforms continue to replace other applications, email has seen steady growth. Thisindicates that advances in secure email will have far-reaching and long-lasting benefits.Because emailtransmission is not secure, many organizations resort to the use of separate securemessaging websites for sensitive communication. When a user receives a message onthese systems, they also receive an email that tells them to log into the secure messagingwebsite to read their sensitive message. This is an annoying and cumbersome process

that is largely disliked by users [1,2]. Secure email would remove the need for theseseparate web applications, allowing sensitive information to be sent directly throughemail.

2. E-Mail Security

When email was first designed in 1971no meaningful attention was paid to security[3]. Securing an e-mail system is the responsibility of an organization's IT department and e-mail administrator. However, anyone responsible for the confidentiality, integrity, and availability of the information sent via e-mail should be aware of the threats facing e-mail systems and understand the basic techniques for securing these systems. Following are the basic requirements for e-mail security:

- **confidentiality**: In addition to privacy, a third party cannot even know whether there was an email or not.
- **Authentication**: The receiver knows the identity of the sender.
- **Integrity**: The receiver knows that the message has not been altered after leaving the control of the sender.
- **Non-repudiation**: The receiver can prove to a third party that the sender really did send this message.

Assuch, it is unsurprising that it was trivial for attackers to steal email and also to inject false messages.As shown in the figure 1, An attacker is able to steal email in five different locations: during transmissionbetween the user and their email server (A, E), during transmission between email servers(C), while at rest on either user's email server (B, D).An attacker is also able to inject false messages at any of the five locations(A, B, C, D, E).

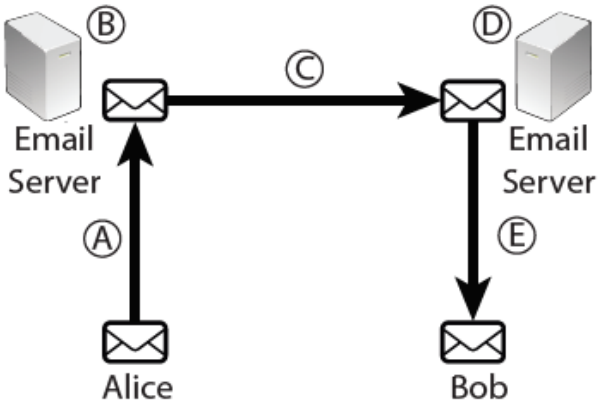

Figure 1 Basic Email Security

Since then, there have beenattempts to patch security on top of email, buteven with these advances it is still possible for an attacker to steal and inject email messages.

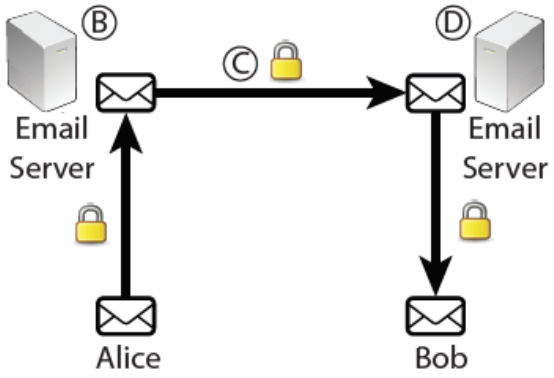

Figure 2Email Security with TLS

As shown in figure 2, by using Transport Layer Security(TLS), an attacker is unable to steal messages during transmission between theuser and their email server. Theoretically, an attacker should not be able to steal messagesduring transmission between email servers (C), but in practice this location is still vulnerableto attackers [10, 12, 18]. As TLS only protects data during transmission, an attacker canstill steal a user's email while at rest at the user's email server (B, D).

Additionally, TLS does not protect against an attacker injecting false messages at anyof the above locations (B, C, D). DKIM can partially protect against message injection duringtransmission of messages between email servers (C), but cannot prevent message injection at the user's email server (B, D).

As such, email is an easy target for attackers. For example, Durumeric et al. foundthat in seven countries over 20% of inbound Gmail messages are being stolen [4]. Also,email can also be stolen while it is stored at the user's email server.8 Additionally, theinability to authenticate the sender of an email increases the likelihood of email phishing, amulti-billion-dollar problem.

3. Cryptography for E-mail Security

Email security refers to the collective measures used to secure the access and content of an email account or service. It allows an individual or organization to protect the overall access to one or more email addresses/accounts.An email service provider implements email security to secure subscriber email accounts and data from hackers - at rest and in transit.Email is the most common threat vector used by cyber criminals. Deploy the cloud-based service that protects your organization from advanced email threats such as targeted phishing attacks, ransomware, business email compromise (BEC) and email fraud.

A cryptographic e-mail security solution reduces the risks associated with regulatory violations, data loss and corporate policy violations while enabling essential business communications. The solution should work for any organization that needs to protect sensitive data, while still making it readily available to affiliates, business partners and users—on both desktops and mobile devices.

By using cryptographic Algorithm we can have following benefits:

- Protect against targeted phishing attacks and email fraud
- Secure your Exchange Online, Gmail and on-prem email servers
- Keep security up to date with real-time threat intelligence
- Enforce strong email data loss prevention (DLP) & compliance
- Get the scalability you need with no upfront costs

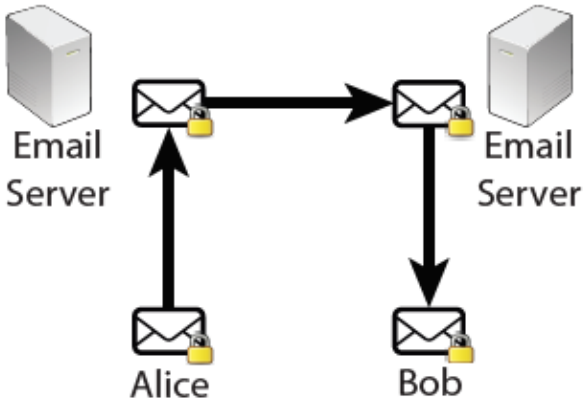

Figure 3 Email Security with End-to-End Encryption

As shown in figure 3, In end-to-end encryption, messages are encrypted at Alice's computer and aredecrypted on Bob's computer. Even if an attacker were able to steal a user's email duringtransmission or storage at an email server, the attacker would be unable to obtain theplaintext content of the message.

In addition to encrypting the email she authors, Alice will also sign it. This allows Bobto verify that the email he received really was authored by Alice. This feature of end-to-endencryption prevents an attacker from injecting false messages.

3.1 Confidentiality

Privacy is threatened by snooping, i.e. eavesdropping on a line, a forensics examination of the sender's, the receiver's, or a MTA's storage system, a monitoring tool on one of the machines involved, etc. A simple way to provide privacy is encrypting the message. This becomes complicated once we send messages to more than a single recipient, unless we want to encrypt all messages. If the number of recipients is small, then the following scheme works:

- Use a secret key K to encode the message.
- Piggy-back $E_U(K)$ onto the message for all recipients, where U is the secret key that the sender shares with a recipient.

The sender can use the following strategy with a remote exploder. Assume that the remote exploder has a shared key with all recipients. Then the sender needs one shared key with the exploder to send the message to the exploder, who then re encodes it with the recipients' secret key. Of course, this will not work if an attacker can add names to the distribution list.

3.2 Authentication

Authentication with public keys is straightforward, the sender signs the message, typically by encrypting the complete message with her secret key. As an alternative, she can use a cryptographically secure hash, that she then encrypts.

Symmetric key technology presupposes a shared key. The sender appends a **Message Authentication Code** (MAC) to the message. Possibilities include the CBC residue of the message computed with the shared secret key, or again the encryption of a secure message digest. The last saves computations for multiple recipients.

3.3 Message Integrity

Typically, message integrity and source authentication are provided with the same tool. It is interesting to ask whether it is possible to guarantee message integrity without source authentication, so this seems to be mostly a concern of the NSA or the criminal underground.

To provide message integrity, we can encrypt the complete message with the public key of the receiver. If the receiver can decode the message and it is an appropriate text, then it had to survived the transmission as a whole. Since the public key is freely available, no conclusions on the source can be drawn.

3.4 Non-Repudiation

Alice repudiates a message if she denies that she ever sent it. Non-repudiation is the property that denies this act to Alice or any other sender of email. This property is different from message integrity with source authentication. For example, operatives in the INTEL community need to be sure of the identity of the sender, but the sender wants the ability to repudiate the message. Here is one way for Alice to send a message to Bob that is authenticated but that she can repudiate.

- Alice invents a secret key S.
- Alice encrypts S with Bob's public key B_{pub} to obtain $B_{pub}(S)$.

- She now signs $B_{pub}(S)$ with her private key A_{priv} and obtains $A_{priv}(B_{pub}(S))$.
- She uses S to compute a MAC for her message.
- She sends the MAC, $A_{priv}(B_{pub}(S))$, and the message to Bob.

Bob uses Alice's public key and then his private key to recover *S*. Bob then verifies that the MAC sent by Alice is indeed the MAC using *S*. Since only Alice knows her private key, this proves that Alice sent the mail. Bob has enough knowledge to fake a message from Alice, as long as he uses the same *S* to construct the MAC. Thus, the only thing that Bob can prove is that Alice sent him an email.

If Alice desires non-repudiability of the email, then she simply encodes the whole email with her private key. Only one with her private key can have issued this message.

If Alice and Bob use secret keys, then they will need a **notary** *N* that must be trusted by Bob and by any judge who is to rule on who is the sender of the message. The notary uses a secret key *S* to **seal messages**. If Alice needs to send something to Bob, she first sends the message to the notary who seals it, and then returns it to her. The notary shares a key with Bob and uses this key to generate a MAC of the message and the seal, that is also appended to the message, which is returned to Alice. She then forwards this message with the seal and the MAC to Bob. Bob can use the MAC to verify that the notary sealed the message. The notary can also determine that this message is genuine and submit this ruling to the judge.

Conclusion

Many believe that it is safe to send confidential information on a company's e-mail. Only few thinks about the fact that in reality e-mail messaging is as open as sending a postcard. Roughly said, anyone could just flip the card over and read what the card says. Ensuring the confidentiality of electronic communication, however, can be simple and easy. In this paper we have discussed some cryptographic techniques tips on how you can ensure the security of an e-mail and other electronic communication where applicable.

References

[1] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy Van Der Horst, and Kent Seamons.Confused Johnny: When automatic encryption leads to confusion and mistakes. In NinthSymposium on Usable Privacy and Security (SOUPS 2013), Newcastle, United Kingdom,2013. ACM.

[2] Scott Ruoti, J. Andersen, Scott Heidbrink, Mark O'Neill, ElhamVaziripour, JustinWu, Daniel Zappala, and Kent Seamons. A usability study of four secure email toolsusing paired participants, 2016.

[3] http://openmap:bbn:com/~tomlinso/ray/firstemailframe:html

[4] S. L. Garfinkel and R. C. Miller, "Johnny 2: A user test of keycontinuity management with S/MIME and Outlook Express,"in First Symposium on Usable Privacy and Security Pitsburg, PA: ACM, 2005, pp. 13–24.

[5] Whitten, A., and Tygar, J. Why Johnny can'tencrypt: A usability evaluation of PGP 5.0. InEighth USENIX Security Symposium(USENIX Security 1999), USENIXAssociation (Washington, D.C., 1999), 14–28.

[6] Sheng, S., Broderick, L., Koranda, C., andHyland, J. Why Johnny still can't encrypt:Evaluating the usability of email encryptionsoftware. In Poster Session at the SymposiumOn Usable Privacy and Security (Pitsburg,PA, 2006).

[7] Ruoti, S., Andersen, J., Zappala, D., andSeamons, K. Why Johnny still, still can'tencrypt: Evaluating the usability of a modernPGP client, 2015. arXiv preprintarXiv:1510.08555.

[8] Garfinkel, S. L., and Miller, R. C. Johnny 2: Auser test of key continuity management withS/MIME and Outlook Express. In FirstSymposium on Usable Privacy and Security, ACM (Pitsburg, PA, 2005),13–24.

[9] Fahl, S., Harbach, M., Muders, T., Smith, M.,and Sander, U. Helping Johnny 2.0 to encrypthis Facebook conversations. In EighthSymposium on Usable Privacy and Security, ACM (Washington, D.C., 2012).

[10] Ruoti, S., Kim, N., Burgon, B., Van Der Horst,T., and Seamons, K. Confused Johnny: Whenautomatic encryption leads to confusion andmistakes. In Ninth Symposium on UsablePrivacy and Security .ACM(Newcastle, United Kingdom, 2013).

[11] ZinaidaBenenson, Gabriele Lenzini, Daniela Oliveira, Simon Parkin, and Sven Uebelacker.Maybe poor Johnny really cannot encrypt: The case for a complexity theory for usablesecurity. In Twenty-Third New Security Paradigms Workshop (NSPW 2015), pages85-99, Twente, The Netherlands, 2015. ACM.

[12] Erinn Atwater, CecyliaBocovich, UrsHengartner, Ed Lank, and Ian Goldberg. LeadingJohnny to water: Designing for usability and trust. In Eleventh Symposium On UsablePrivacy and Security, pages 69{88, Montreal, Canada, 2015. USENIXAssociation.

[13] SaschaFahl, Marian Harbach, Thomas Muders, Matthew Smith, and Uwe Sander.Helping Johnny 2.0 to encrypt his Facebook conversations.In Eighth Symposium onUsable Privacy and Security, page 11, Washington, D.C., 2012. ACM.

[14] Matthew D Green and Ian Miers. Forward secure asynchronous messaging frompuncturable encryption. In Thirty-Sixth IEEE Symposium on Security and Privacy(S&P 2015), pages 305{320, San Jose, CA, 2015.IEEE Computer Society.