

Secure Email Protocol Comparison: PGP and S/MIME.

Karam Hack

ka265070@ucf.edu

University of Central Florida

Orlando, Florida, USA

ABSTRACT

Until now, email security remains a critical challenge. Traditional communication protocols were never designed with privacy in mind. This project explores and compares two major secure email protocols; Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME), focusing on their usability, cryptographic guarantees, and adoption barriers. The project aims to analyze how each protocol approaches encryption, authentication, and key management, while also evaluating their effectiveness and limitations in modern use. With literature review, technical analysis, and diagrammatic explanations, the project highlights both the theoretical and practical aspects of secure email. The expected outcome highly expresses that, despite the cryptographic heaviness, both PGP and S/MIME face significant usability and scalability challenges that limit their widespread adoption. This will offer insight into how future frameworks can balance usability with strong cryptographic expectations/assurances.

Github Repository containing a directory of references:

[Click Here to access the Repository](#)

1 INTRODUCTION & PROBLEM STATEMENT

With our current ever evolving technology, we still run the risk of downsides with using protocols/standards. We have heard traditional email protocols such as SMTP, POP3, and IMAP that are widely used but were not originally designed with the privacy and security in mind. When the internet was first designed, it was never made with the intention of security, and this cause has effected our time today with our modern tools. To address the concerns, Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extension (S/MIME) have the same goal in mind but different cryptic methodologies. Both aim to provide end-to-end protection through encryption, authentication, and digital signatures, yet they employ fundamentally different cryptographic models and trust mechanisms.

Despite their technical strengths, these protocols face persistent challenges in usability, scalability, and adoption. PGP's decentralization with *Web of Trust* lends user control but lacks simplicity for non-technical users, while S/MIME's reliance on centralized Certificate Authorities (CAs) introduce cost, complexity, and general overhead.

2 RELATED WORK

The research accumulates with a mixture of findings from scholarly articles, technical analysis, and insights from other research that goes hand in hand with usability. Here is the following relative experience that contributes to the depth of this paper:

- **In-Depth Comparison:** A comparison between both protocols (PGP and S/MIME) in regards to usability, security and reliability while also discussing the weaknesses that can compromise them. By first taking a look into the background of each protocol and how they have matured in modern time.
- **Discoverable Research:** There are comparative analysis out there to look into. That can relate the depth of the comparison of both protocols while also addressing the indicators for their security.
- **Usability Cases:** The general questions of "How can I use this?" or "How does this generally work?" while also signifying on what could improve their functions

3 TECHNIQUES & METHODOLOGIES

The study employs a comparative analysis of the two most dominant secure mail protocols, Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME). The research has a historical analysis based on development and design intentions as aimed at discerning how the respective platform has progressed over time in response to the absence of privacy in the traditional mail programs. The study shall compare the two protocols in usability, cryptography assurances, in addition to barriers to entry, based on contemporary literature and technical descriptions.

In order to provide an organized comparison, individual protocols will be discussed separately, their encryption and authentication procedures, model of keys, and workflow of operations explained in detail. Diagrammatic visuals will be provided to explain the process of encryption and trust models of the protocols. Once the base has been explained, the deficiencies and frailties experienced in practical applications shall be discussed, drawing case studies and empirical evidence to compare their usability in contemporary settings.

4 EVALUATION & RESULTS

4.1 PGP History

The creator of PGP, Phil Zimmermann, started to write PGP in the mid 1980s and finished the first version in 1991. It started to become public as free software, but issues appeared by noticing that Zimmermann's release had no rights in releasing the cryptographic method that worked hand in hand with PGP: the public key cryptography patents. This led to PGP being called bandit-ware instead of freeware. The release was due to a hasty decision on Zimmermann's end based on financial struggles, legal struggles with RSADSI and a newly sparked Anti-crime law (S.266) explaining that anything is requested by law of plain text contents of voice, data, and other communications that are requested by law. [5]

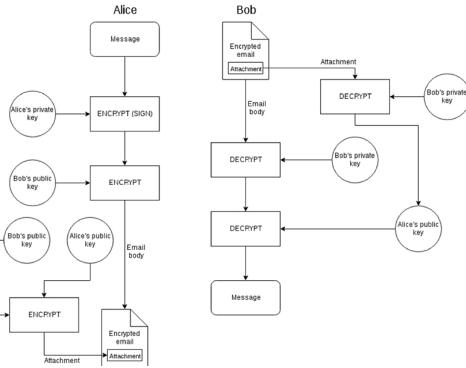


Figure 1: PGP Process [11]

Due to the legal atmosphere, it attracted Public Key Partners (RSADSI and Cylink, which are the holders of Diffie-Hellman and Hellman-Merkle patents) in announcing to other organizations that the use of PGP should be highly discouraged and should not be used for anything. But Zimmermann believed that encryption was a way to protect democratic freedoms in a networking world that's in pace to increase at an exponential result. Not only was PGP first in strong cryptographic methods but it was also used in simple command-line usability. [5][11]

Zimmermann's created his own algorithm for encryption, specifically symmetric algorithm, called *Bass-O-Matic*, paired with RSA public-key functions for encryption and digital signatures. Later down the line, Zimmermann's was convinced by another by the name of Eli Biham, a cryptographer from Weizmann, that *Bass-O-Matic* contained many rookie mistakes when it came to encrypting. For example, Zimmerman's algorithm prevented the last bit of each byte from being properly encrypted. This led *Bass-O-Matic* being thrown out the window and bring out a new algorithm, International Data Encryption Algorithm (IDEA). IDEA provided a 128-bit symmetric cipher regarded as substantially stronger and more efficient. Presenting a hybrid approach, PGP randomly generates a session key encrypted with the recipient's RSA public key, while the data is encrypted with IDEA. A hashing algorithm (MD5) supplied message-digest functions for verification, while the digital signatures came and was authenticated from senders and ensured non-repudiation (proving origin of data). [Figure 1, 11]

PGP had a distinctive feature and that is its decentralized trust model, which is known as Web of Trust. Instead of an authority that validates keys, users take the authority themselves. They will take each others keys and personally sign it for distributed credibility. This aligned with Zimmermann ideological aim of getting rid of institutions having the ability to mandate and control over all privacy tools. Yet, the usability barriers that came to be is verifying key fingerprints manually, management of revocation lists, and gaining complex insight on how to operate the signature chains well beyond the average of most email users based on technical proficiency.

Throughout the majority of 1990s, the evolution that took place for PGP came from the "guerrilla freeware" of Version 1.0 to the standardized format used today, OpenPGP. Due to its cryptographic

architecture, by providing new means of confidentiality and integrity. It's usability and key-management complexity restricts adoption outside the field of experts.

4.2 S/MIME History & Encryption

Within the 1990s is when Secure/Multipurpose Internet Mail extensions (S/MIME) emerged. When PGP was hitting legal action due to Anti-crime Law and patent laws. This was the legal industry-standard framework for secure email, which was developed to integrate encryption and digital signatures directly from a pre-existing format, which was Multipurpose Internet Mail Extensions (MIME). This format was previously used by most Internet mail systems. Whereas PGP arose to initiate to decentralize the trust. S/MIME originated within large corporations and government ecosystems that relied on centralized email security. This traces back to Privacy Enhanced Mail (PEM), which is an earlier effort by the Internet Engineering Task Force (IETF) within the late 80s to secure with a certificate called X.509 and a hierarchical Public Key Infrastructure (PKI). Due to the complexity and rigid trust of PEM's resulted in the prevention of widespread adoption, but it provided the blueprints to setup the new foundation for S/MIME.

During 1995, a collaboration between RSA Data Security Inc. and industry partners such as Lotus and Microsoft created the specifications for S/MIME. It used CMS (Cryptographic Message Syntax), which is from ISO/ITU-T PKCS #7 standard. Its purpose is to create private signed or encrypted data within the MIME framework. This allows messages, specifically secured messages to exist normally within email traffic by not altering the SMTP (Simple Mail Transfer Protocol) transport. S/MIME contains RSA for key exchange and signatures, DES (Data Encryption Standard) or 3DES (Triple DES) for symmetric encryption, and SHA-1/SHA-2 for the use of hashing.

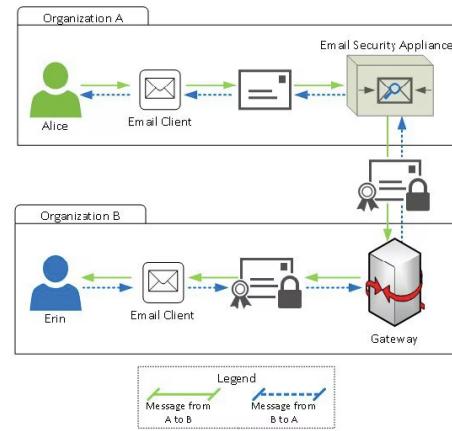


Figure 2: S/MIME process Business-to-Consumer [12]

S/MIME's whole design framework is to work with X.509 certificate standard. Each user's public key is bound to an identity based on the Certificate Authority (CA). Which forms a chain of trust. This allows a better revocation list, providing centralized control

and enforcement of policies. However, if a CA is compromised, the entire chain of trust gets compromised or invalidated. [Figure 2, 12]

Under the usability parameter, S/MIME was also designed to be transparent for end users in managed environments. Most popular email clients, automatically retrieve certificates from corporate directories (CD) or LDAP (Lightweight directory access protocol) servers. Yet the process of getting certificates outside of this framework is complex. The reason, is that most certificates usually require some form of payment or affiliation to the institution and mechanisms for revocation such as CRLs (Certificate Revocation Lists) and Online Certificate Status Protocol (OCSP). Due to the complexities, hinders the widespread adoption of S/MIME within the consumer realm.

S/MIME covered all bases when it comes to the CIA triad, which is supported by widely standardized algorithms and formal PKI validation procedures. Unlike PGP decentralization, S/MIME's security relies on the integrity of Certificate Authorities and certification validation protocols from email clients. This model offers the stability and enforcement of policy but lacks transparency and autonomy. Usability studies consistently indicate that users rarely comprehend certificate warnings or errors, which causes a override of security prompts and weaker protection.

4.3 OpenPGP/PGP Encryption

As mentioned before, PGP has a hybrid cryptographic method that combines the **symmetric key** and **public key cryptography techniques**. The way that symmetric key cryptography works is that there will be one secret key for both the encryption and decryption of data and in for public key cryptography uses two keys: which is a public key that is shared with everyone for the encryption, and a private key that is kept in secret for decryption. [1][2][3][5][13][11]

As always, there is a form of authentication in the email world. This is a requirement so that people don't spoof emails or spam. Starting from the sender: [1][2][3][5][13][11]

- A hash function (SHA-1) will generate a 160-bit hash value of the said message.
- The hash is encrypted with the senders private key which makes a digital signature.
- The message and the signature will then combine, then compress and be sent to the receiver.

During the receiving end:

- The message will then get decompressed.
- The signature is decrypted using the senders public key (PUa) to retrieve the hash.
- The message will then be hashed again, using the same function since the beginning .
- If both of the hash values match, then the message is verified as an authentic and unaltered message. Otherwise, most likely the message is from an untrusted end point.

For confidentiality purposes:

- PGP provides secrecy by using random session keys with symmetric encryption (**CAST-128, IDEA, or 3DES**).
- The session key will be encrypted using the receivers public key with the use of **RSA encryption**.

- Then the encrypted message and encrypted session key will be sent altogether to the receiver at the receiving end of this process.

During the receiving end:

- The session key will be decrypted using the receivers private key.
- When the message is encrypted using the retrievers session key, the messages then decompresses to restore the original content.
- If the session key that was encrypted and got intercepted, it will still remain unreadable without the use of the private key.

PGP combines the concepts of confidentiality, authentication and integrity (CIA) together as one.

4.4 PGP Advantages & Disadvantages

The one thing that PGP is primarily known for is its algorithm, unbreakable algorithm second of all, it is regarded as a top technique for improving the cloud security space and will also be frequently utilized by user who needs to encrypt their private conversation conversations. PGP has the ability to prevent hackers from accessing any government files or emails are encrypted with PGP, despite the recent upcoming of vulnerabilities that I will mention down the line.

One of the main drawbacks of PGP encryption is that it's very complex. It has a high learning curve which consumes time to fully encrypt data and files, which makes it even more difficult for new users so if an organization is wanting to have PGP to be utilized, it will then cost the company or organization to train their employees. Along with the learning curve users have to understand the specifics of the PGP system to prevent unintentionally weakening the security measures this range from PGP being used correctly or losing or corrupting keys while also endangering others in situations where security is at an extreme. Although PGP has really good encryption, it does not make the messenger anonymous. PGP also will not work unless both the sender and recipients are using the same version of the PGP software. [1][2][3][5][13][11]

4.5 S/MIME Advantages & Disadvantages

Since we delve into the advantages and disadvantages for PGP, we will now discuss those pros and cons for S/MIME. Starting off with the advantages, S/MIME is a very secure email encryption just like PGP it offers digital integrity and privacy. That helps prevent attacks like phishing and enable secure communication. S/MIME will be able to integrate with multiple different email solutions however some of those solutions don't provide a native S/MIME certificate support. But as a majority, S/MIME is basically supported by almost every common email client and server. S/MIME can also integrate into business environments a lot easier than PGP. S/MIME mail gateways also give admin more variety or scope for company specific configurations. As stated previously, S/MIME history or background, I mentioned that it uses X.509 certificates which are issued by the certificate authority. This alone allows them to enable higher levels of trust certificate authorities, ensure that keys and owners really belong together. This does not come with openPGP. However, the use of certificates is also the limitations of S/MIME

such as the size restriction for file attachments and compatibility issues. For example, if a recipient uses S/MIME and the sender uses openPGP they cannot exchange key pairs as the two of them have different methods when it comes to sending and receiving. [1][2][3][5][13][11]

Since S/MIME has asymmetric key pairs. Those key pairs have a very long lifetime. If a third-party were to have access to the communication and a private key, not only will they be able to see the email but all the data that can be decrypted with that said compromised key. When it comes to messages being protected by encryption during transmission, other metadata, such as the subject line will remain unencrypted. S/MIME can interfere with other security protocols, such as antivirus, scanners archiving tools, and other data loss prevention mechanisms. For example, Emails are frequently scanned for viruses on the way out of the network but because the email is encrypted, S/MIME will hide the contents from the receiving gateway scanner, which means anything can bypass a firewall.

Just as PGP, S/MIME can also be very costly. It has commercial certificates that must be purchased from a trusted certificate authority, so the investment is kept in mind. [1][2][3][5][11][13]

4.6 Vulnerabilities of PGP and S/MIME

So a vulnerability that is quite highlighted within the PGP and S/MIME protocol is the encryption modes that are implemented. For example, S/MIME, uses something called Cipher Block Chaining (CBC) and OpenPGP uses Cipher Feedback Mode (CFB) and while within the description process, both of these modes produce a value, which are XORed within an adjacent ciphertext block to produce the final plain text block, knowing this it also holds a limitation behind that because not only do they have precise modifications of the plain text, but they can also re-order around the plain text blocks and an extension to the limitation changing those blocks or re-ordering them can allow an attacker to reflect that in the URL path and the resulting HTTP request and ex-filtrate sensitive data.

Exfiltration attacks again can be applied to OpenPGP and S/MIME however each have different obstacles, but in general, they can still be affected by these exfiltration techniques (EFAIL). Another way of attacking both of these protocols is something called Signature Spoofing Attack. This usually means that you're able to make a fake signature verification and let the receiver believe that they got a message from the sender based on the signature. The reason why OpenPGP is susceptible to signature spoofing is because of the lack of UI design and some of the clients. Even S/MIME is more automated but, it is also criticized for when certificates from the certificate authority (CA) are expired, untrusted or revoked; which gives a false sense of integrity or security. [1][2][3][5][11][13][14][15]

5 DISCUSSIONS & CHALLENGES

5.1 Discussion Questions

I will adopt the systematic literature review (SLR) as a means to asking discussion questions of the research comparison between PGP and S/MIME. [13]

The questions are as follows:

- (1) What are the differences between PGP and S/MIME?

#	PGP	S/MIME
1	Less support for webmail	not available for webmail clients
2	requires third party plugins	works natively with enterprises
3	key pairings are set up manually	CA management
4	Less compatibility	Problems with cross-device compatibility

Figure 3: A table comparison of Integrations for PGP and S/MIME [1][2][3][5][11][13].

#	PGP	S/MIME
1	Uses a hybrid approach for message encryption.	Non-hybrid approach under the PKCS7
2	User controlled when it comes to keypair management	Contains automation based on client or policy settings
3	Heavy learning curve but offers transparency	Certificates are managed by CA
4	More configurations and flexibility within workflows	Easy to use but lacks transparency
5	Selection of encryption and signing types	Suited for corporate/enterprise deployments

Figure 4: A table comparison of mechanisms and usability for PGP and S/MIME [1][2][13]

- (2) How does the cryptographic framework influence the security and usability of the two protocols?
- (3) What are vulnerabilities/threat models discovered in regards to PGP and S/MIME?
- (4) What are the best case scenario use for the protocols?
- (5) How are certificates and key revocations handled?
- (6) Describe mitigations can be utilized for the discovered vulnerabilities?
- (7) What integration challenges occur in modern platforms?

5.1.1 What are the differences between PGP and S/MIME?

One of the key insights looking into this research is the fundamental difference between PGP and S/MINE and the approach to trust. PGP utilizes a decentralized architecture through the web of trust, where users were just validate and endorse each other however, this all comes with the price of a learning curve. With S/MIME operating on a public key infrastructure (PKI) model and leveraging a certificate authority to basically validate, identify and revoke digital certificates. Due to the third-party of certificate, authorities can create systemic vulnerabilities if ever compromise. [1][2][3][5][11][13].

5.1.2 How does the cryptographic framework influence the security and usability of the two protocols?

A significant key point inside into the cryptographic mechanisms that influence the security and use ability of the two protocols is that they both use cryptographic algorithm such as RSA, AES, and ECC. However, PGP will encrypt messages using symmetric keys. And the symmetric key will be encrypted with the recipients public key. This whole process will require the user to manage the keypairs manually and select signing or encryption options, explicitly before sending an email. Hence, the learning curve that I previously mentioned. S/MINE would basically fold or wrap the contents using the PKCS standards

#	PGP	S/MIME
1	Decentralized (Web of Trust)	PKI (Public Key Infrastructure)
2	User-heavy for keys and signing	CA with automated validation
3	High transparency and control	Less transparent but user friendly
4	Prone to user error	Vulnerable to CA mismanagement
5	Suited for user privacy	Suited for enterprise environments

Figure 5: A table comparison of architecture for PGP and S/MIME [1][2][13]

and integrate encryptions in signing them with a MIME structure of the message. The key management itself is largely abstracted relying on the certificates issued by the third-party certificate authorities. This allows S/MIME more available to non-technical users, particularly in business settings, but reduces the visibility and manual control over the operations.[1][2][3][5][11][13].

5.1.3 What are vulnerabilities/threat models discovered in regards to PGP and S/MIME?

The discovered vulnerabilities for both the protocols is something called EFAIL. A very notable attack which exploits the flaws and how the email clients render HTML and handle encrypted MIME content. This attack basically allows partial plain text exfiltration via crafted message structure. PGP will also suffer from signature spoofing, which primarily comes from the structure of the verification. S/MIME, can silently fail when certificates basically expire or lose trust or get taken away, giving users, a false sense of integrity.[14][15].

5.1.4 What are the best case scenario use for the protocols?

In short, PGP is generally looked upon when it comes to privacy conscious environments were users wants to value control over trusted relationships. Keen on flexibility and independent away from central authorities. S/MIME is purely designed with the enterprise in mind. It will integrate easily with major email clients and support a centralized policy enforcement. It also has very adequate scaling across large organizations.[1][2][3][5][11][13].

5.1.5 How are certificates and key revocations handled?

The concept of revocation is vital for maintaining trust within the secure communication systems. PGP supports this process with user generated certificates, which must be created in advanced and uploaded to public key servers. While this does give full control to the user. It also introduces risk if not generated or published in a timely manner. S/MIME will rely on certificate, revocation mechanisms, such as Certificate Revocation List (CRL) and the Online Certificate Status Protocol (OCSP). These two work hand and hand by checking a real time list, but the effectiveness is heavily reliant on the enforcement and implementation of clients.[1][2][3][5][11][13].

5.1.6 Describe mitigations can be utilized for the discovered vulnerabilities?

A mitigation to exfiltration attacks is a same origin policy for an

#	PGP	S/MIME
1	Private communities and personal use	Enterprises and corporate environments
2	supports web of trust and independence	Enforces CA and organization policies
3	popular in flexible fields	Accepted in the government systems
4	background knowledge needed for setup	certificate handling automated and easier deployment
5	Difficult to scale	Easy integration and scalable

Figure 6: A table comparison of integration for PGP and S/MIME [13][16][17]

emails. Since HTML, CSS and MIME makes it possible to mix encrypted and plaintext contents. However, the mitigation itself is hard to enforce in every scenario in mind. The reason for this is because the email gateways in companies process the encrypted emails and then forward the plain data to email clients used by the employees. The clients will have no knowledge whatsoever whether the original message was encrypted or not so would this counter measure in mind and must be combined with different techniques. In regards to signature spoofing, PGP must track attacker, control data and always escape, new lines and other special characters and all outputs. Front end developers can harden the indication process of the backend. Open PGP should only one optional encryption layer, one optional compression layer, and one possible signed literal data packet. They should be kept in mind for standard revisions for the future. For S/MIME, there are two distinct ways in mitigating this. One of them is, emailing clients should show how exactly part of the message was signed or email. Client should be conservative and only showing a message as correctly signed if the whole message was correctly signed another important factor is to show time stamps so if a timestamp is older than a certain threshold, it can give out an alert which protects attacks from a whole life time of a certificate.[14][15].

5.1.7 What integration challenges occur in modern platforms?

The main hurdle when it comes to integration challenges in the modern platforms is mobile clients and web mail. The reason is, because PGP requires third-party plug-ins, browser extensions, and some external apps to properly mechanize within popular platforms like Gmail. S/MIME is better when it comes to integrating, however, it does not support browser based email clients and that poses challenges in zero trust or Bring Your Own Device (BYOD) scenarios and additionally, the dependence on local certificates can give a rise to compatibility issues across many devices.[1][2][3][5][11][13].

5.2 Challenges

Throughout the journey of doing this research, one of the challenges facing this is truly finding the depth and the time to accommodate searching that insight for the protocols. Initially not looking into the right places when it comes to gathering information, but once I've accessed certain databases, it really helped a lot. Putting the pieces together and how I should format the results needed for this comparative analysis project was another stepping stone.

6 CONCLUDING REMARKS

This is a deep dive analysis comparing two secure email protocols PGP and S/MIME. I evaluated them across seven questions based on a research benchmark by talking about the architecture, the mechanisms, the usability, vulnerabilities, integration context and ways to mitigate the vulnerabilities. Research revealed that protocols alone are very secure. However, it depends on the usability environment and what the client needs when it comes to the quality of security. PGP will provide users, transparency, and more control but suffer a learning curve. S/MIME is quite the opposite. It's dedicated for enterprises, and it is centralized to a certificate authority, while being automated.

Research vulnerabilities such as signature spoofing and EFAIL continue to persist in both protocols integration with modern platforms like Webmail or Mobile-mail remains a hurdle to solve. Mitigations mentioned in regards to the vulnerabilities is enforcing Same Origin Policy for emails, tracking attackers controlling the data, fortifying processes in the backend for signature spoofing.

As a result, it has come to a discovery that we must find a paradigm or an architecture for encryption to be user-friendly and compatible across many ecosystems that are both personal and enterprise demand.

7 DEDICATION

I did not use AI tools to write this report.

REFERENCES

- [1] S/MIME vs. PGP - What is the difference?
- [2] S/MIME vs. PGP
- [3] Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels
- [4] S/MIME vs PGP by Securitybuddy
- [5] Garfinkel, Simson. PGP: Pretty Good Privacy. 1st ed. Sebastopol, CA: O'Reilly & Associates, 1995. Print.
- [6] Reuter A, Abdelmaksoud A, Boudaoud K, Winckler M. Usability of End-to-End Encryption in E-Mail Communication. Front Big Data. 2021;4:568284. Published 2021 Jul 14. doi:10.3389/fdata.2021.568284
- [7] Kapadia A. "A Case (Study) For Usability in Secure Email Communication" – Focuses on S/MIME (and PGP) usability from a case study perspective.
- [8] Zibran M.F. "Cryptographic Security for Emails: A Focus on S/MIME" – A detailed review of S/MIME architecture with usability and adoption commentary.
- [9] N. Freed and N. Borenstein. 1996. RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC Editor, USA.
- [10] N. Borenstein and N. Freed. 1993. RFC1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. RFC Editor, USA.
- [11] PGP Encryption Software: What is it and How Does it Work? (Fig.1)
- [12] S/MIME Security Services in Email Gateway (Fig.2)
- [13] A Comparative Evaluation of Secure Email Protocols: PGP vs. S/MIME

[14] Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels
Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels

[15] "Johnny, you are fired!" -Spoofing OpenPGP and S/MIME Signatures in Emails "Johnny, you are fired!" -Spoofing OpenPGP and S/MIME Signatures in Emails

[16] S/MIME encryption: who needs it & how to get it

[17] S/MIME vs. OpenPGP | What's the difference?