From Expired Certificates to Identity Mismatches: How Well Do Email Clients Communicate S/MIME Issues?

Mario Gemov
Karlsruhe Institute of Technology

Fabian Ballreich
SECUSO, Karlsruhe Institute of Technology

Melanie Volkamer
SECUSO, Karlsruhe Institute of Technology

Abstract

This research explores how modern email clients handle security and technical issues associated with S/MIME encrypted and signed emails. By testing 13 widely used clients across multiple platforms, it examines how well they detect common issues - such as expired or revoked certificates and mismatched sender identities - and how clearly these problems are communicated to users. The results show noticeable differences in detection accuracy: while some clients consistently performed well, others - particularly Android-only apps and mobile versions - proved less reliable. Warning clarity also varied, with some clients providing misleading information. Overall, the findings show that even though S/MIME has been available for many years, many email clients still struggle to properly handle and communicate related security issues.

1 Introduction

Secure communication via email remains a critical requirement in both personal and professional contexts. One widely adopted standard supporting such security is the Secure Multipurpose Internet Mail Extension (S/MIME), which provides mechanisms for encryption and digital signatures.

Although S/MIME is built on strong cryptographic principles, its practical effectiveness depends on how email clients implement and enforce its features. Inconsistent or incomplete handling of certificate and signature issues can result in unclear or missing security warnings for users.

A 2009 study examined how email clients handle mismatched email addresses in signed emails [5]. This research extends their work by investigating how email clients handle various S/MIME-related issues, with a focus on detection and communication of certificate, signature, and decryption problems. The main results show substantial variation between clients: a small number consistently detect and clearly report all tested issues (Figure 5), while many others fail to identify certain problems or provide unclear or misleading warnings (Figure 6). Mobile clients perform less reliably than desktop clients, both in detection coverage and warning clarity.

A cross-platform set of widely used email clients was evaluated using various S/MIME test scenarios. Client behavior was systematically recorded and compared.

2 Methodology

Based on S/MIME functionality and requirements, a set of test scenarios was defined, that reflect realistic situations involving technical and security issues. This includes (1) certificate expiration, (2) certificate revocation, (3) invalid signatures, (4) untrusted certificate authorities, (5) identity mismatches, and (6) missing decryption keys. These scenarios were inspired by an illustration of the life cycle of a signed message from [5].

A total of 13 email clients were chosen based on their widespread use, availability, and support for S/MIME. Figure 3 lists the tested clients and versions. Initially based on the clients listed in [9]. The testing was conducted across the most popular desktop and mobile platforms: Windows, Linux, macOS, Android, and iOS. [10]

Each test scenario was implemented either by using specific certificates or by modifying the corresponding EML files. For most scenarios, certificates provided by the Karlsruhe Institute of Technology (KIT) were used. At the time of testing, these certificates were issued via the GÉANT TCS service, with Sectigo acting as the underlying certificate authority. [4] Self-signed certificates generated via OpenSSL were used only for cases requiring specific configurations [2]. This was the case with the expired certificate scenario, where a certificate with a one-day validity period was created. The revoked certificate scenario was set up using a KIT certificate revoked through

USENIX Symposium on Usable Privacy and Security (SOUPS) 2026.
August 23–26, 2026, Hannover, Germany.

the Sectigo Certificate Manager to simulate real-world revocation. The invalid signature scenario was created by modifying the email body in the EML file after signing to break the hash match. The non-trusted Certificate Authority scenario used a self-signed certificate issued by a Certificate Authority, which was later not made known to the clients. The identity mismatch scenario was implemented by altering the "From" header in the EML file to differ from the certificate's identity. The missing decryption key scenario was set up by excluding the recipient's private key from the client configuration. Testing emails were imported into the email clients and their responses to these scenarios - warnings and error messages - were recorded for analysis.

3 Evaluation Framework

To ensure a consistent evaluation of email client behavior, a set of categories was defined based on observed differences in how clients presented and handled issue-related information across the test scenarios:

Issue Detection: Whether the issue is detected and acknowledged by the email client through any visual clue.

Warning Visibility: Whether the warning for a given issue appears directly in the main email view or only after additional user interaction.

Information Source: Whether warning information is shown through the email client's own interface or only through a system-level window outside the client.

Color Usage: Whether risk-related colors such as red, orange, or yellow are used to draw the user's attention. [3]

Specificity of Information: Whether the client provides a clear and detailed explanation of the issue or only a generic message.

Misleading Information: Whether the information provided by the client about the issue is incorrect or misleading, such as referring to a different problem than the one being tested.

4 Results and Discussion

Only two of the tested clients detected all issues across all supported platforms. The remaining clients either did not detect some issues at all or showed different detection success depending on their version or the platform on which they were tested. Overall, none of the issues we tested could be detected by all clients. (Figure 1)

Android clients performed worst overall. One client did not detect any signature-related issues and only responded when attempting decryption without a certificate, while another performed slightly better by detecting invalid signatures and providing some guidance regarding missing certificates.

Although the use of risk-related colors has been shown to improve the effectiveness of warning message communication [1], only 54% of the tests featured direct use of such colors (Figure 2). While some clients consistently applied warning colors, many used them only in secondary windows, which reduced their effectiveness in initially capturing the user's attention, or did not use warning colors at all in any of the tested scenarios.

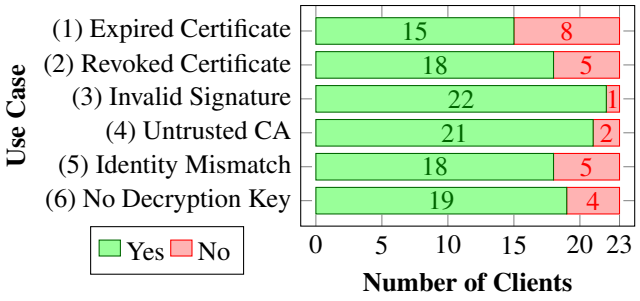

Figure 1: Issue detection across use cases. Each bar represents the number of clients that detected an issue in a specific use case versus those that did not.

Color usage	
Type	**% of tests**
Direct color usage	54%
Indirect color usage	21%
No color usage	25%

Specificity of information	
Type	**% of tests**
Direct specific information	29%
Indirect specific information	68%
No specific information	3%

Figure 2: Summary of warning color usage and specificity of information across all tests.

Regarding the specificity of information, in only 3% of the tests was no specific information provided at all (Figure 2).

Despite their strong detection performance, Thunderbird and SeaMonkey produced misleading warnings by labeling expired or revoked certificates as "untrusted certificate authority". Investigation of their source code revealed that this is a known issue dating back to at least 2008 and has not yet been properly resolved. [6], [7], [8]

5 Conclusion

This work evaluated how well modern email clients handle security and technical issues in S/MIME-encrypted and signed emails. Using different test cases across multiple clients and platforms, it examined issue detection and clarity of user communication. Results show substantial variation: some clients reliably detected all issues, while others failed frequently, particularly on mobile clients. Warning quality also differed, ranging from clear alerts to vague or misleading messages. Overall, S/MIME handling remains inconsistent and needs improvement. This work-in-progress presents a preliminary analysis of S/MIME handling in email clients as part of a larger project. Future work will include a user study on user behavior when verifying signatures.

Acknowledgments

This work was supported by funding from the project "Engineering Secure Systems" of the Helmholtz Association (HGF) [topic 46.23.01 Methods for Engineering Secure Systems] and by KASTEL Security Research Lab.

References

[1] Curt C Braun, Brian Greeno, and N. Clayton Silver. Differences in behavioral compliance as a function of warning color. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 38(5):379–383, 1994. https://journals.sagepub.com/doi/abs/10.1177/154193129403800506.

[2] John Dalesandro. Create self-signed s/mime certificates. https://johndalesandro.com/blog/create-self-signed-smime-certificates/, 2024.

[3] International Organization for Standardization. Iso 3864-2:2016 - graphical symbols – safety colours and safety signs – part 2: Design principles for graphical symbols for use in safety signs. ISO, 2016.

[4] KIT CA. End of gÉant's contract with sectigo. https://docs.ca.kit.edu/geant-tcs/en/sectigo_contract_ends/.

[5] Albert Levi and Can Berk Güder. Understanding the limitations of s/mime digital signatures for e-mails: A gui based approach. *computers & security*, 28(3-4):105–120, 2009. https://www.sciencedirect.com/science/article/pii/S0167404808000783.

[6] Mozilla Project. S/mime security info file. https://hg.mozilla.org/comm-central/file/tip/mail/locales/en-US/chrome/messenger-smime/msgSecurityInfo.properties.

[7] Mozilla Project. msgreadsecurityinfo.js - s/mime message security handler. https://hg.mozilla.org/comm-central/file/e4f4569d451a5e0d12a6aa33ebd916f979dd8faa/mailnews/extensions/smime/resources/content/msgReadSecurityInfo.js, 2008. Changeset e4f4569d451a5e0d12a6aa33ebd916f979dd8faa.

[8] Mozilla Project. msgreadsmimeoverlay.js - s/mime overlay. https://hg.mozilla.org/comm-central/file/17750ab3819d941a11f80085325dfbf1de3757e0/mailnews/extensions/smime/msgReadSMIMEOverlay.js, 2025. Changeset 17750ab3819d941a11f80085325dfbf1de3757e0.

[9] Jens Müller, Marcus Brinkmann, Damian Poddebniak, Hanno Böck, Sebastian Schinzel, Juraj Somorovsky, and Jörg Schwenk. {"Johnny}, you are {fired!"}–spoofing {OpenPGP} and {S/MIME} signatures in emails. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1011–1028, 2019. https://www.usenix.org/system/files/sec19-muller.pdf.

[10] Procurri. Global os market share 2025: Key stats, trends, and insights for mobile and desktop. https://www.procurri.com/knowledge-hub/global-os-market-share-2025-key-stats-trends-and-insights-for-mobile-and-desktop/.

A Appendix

A.1 Email Clients

OS	Client	Version
Windows	eM Client	10.1.4589
	Outlook	1.2025.506.300
	Outlook (Classic)	16.0.18730.20168
	SeaMonkey	2.53.20
	The Bat!	11.4.1
	Thunderbird	128.6.0
Linux	Evolution	3.54.3
	KMail	6.3.1
	SeaMonkey	2.53.20
	Thunderbird	128.6.0
macOS	Apple Mail	16.0
	eM Client	10.1.4828
	MailMate	2.0
	Outlook	16.97
	Outlook (Classic)	16.97
	SeaMonkey	2.53.20
	Thunderbird	128.6.0
Android	eM Client	10.0.3530
	FairEmail	1.2262
	Nine	4.9.5f
	Samsung Email	6.2.02.6
iOS	Apple Mail	17.5.1
	eM Client	10.0.3531

Figure 3: Email clients tested in the thesis

A.2 Issue Detection

Figure 4 summarizes the issue detection data collected for each client. To improve readability, clients available across multiple operating systems that showed no differences in issue detection have been combined into a single row.

A.3 Proper Detection Example

Figure 5 shows how Thunderbird detects a mismatch between the sender and the signer of the email and gives accurate information to the user.

OS	Client	Expired Certificate	Revoked Certificate	Invalid Signature	Untrusted CA	Identity Mismatch	Missing Decryption Key
Windows	eM Client	Yes	Yes	Yes	Yes	Yes	Yes
	Outlook	No	No	Yes	Yes	Yes	Yes
	Outlook (Classic)	Yes	Yes	Yes	Yes	No	No
	SeaMonkey	Yes	Yes	Yes	Yes	Yes	Yes
	The Bat!	Yes	Yes	Yes	Yes	Yes	-
	Thunderbird	Yes	Yes	Yes	Yes	Yes	Yes
Linux	Evolution	No	No	Yes	Yes	Yes	Yes
	KMail	Yes	Yes	Yes	Yes	No	No
	SeaMonkey	Yes	Yes	Yes	Yes	Yes	Yes
	Thunderbird	Yes	Yes	Yes	Yes	Yes	Yes
macOS	Apple Mail	Yes	Yes	Yes	Yes	Yes	Yes
	eM Client	Yes	Yes	Yes	Yes	Yes	Yes
	MailMate	No	Yes	Yes	Yes	Yes	Yes
	Outlook	Yes	Yes	Yes	Yes	Yes	No
	Outlook (Classic)	Yes	Yes	Yes	Yes	Yes	Yes
	SeaMonkey	Yes	Yes	Yes	Yes	Yes	Yes
	Thunderbird	Yes	Yes	Yes	Yes	Yes	Yes
Android	eM Client	No	Yes	Yes	Yes	Yes	Yes
	FairEmail	No	No	Yes	Yes	Yes	Yes
	Nine	No	No	No	No	No	Yes
	Samsung Email	No	No	Yes	No	No	Yes
iOS	Apple Mail	Yes	Yes	Yes	Yes	No	Yes
	eM Client	No	Yes	Yes	Yes	Yes	Yes

Figure 4: Issue Detection

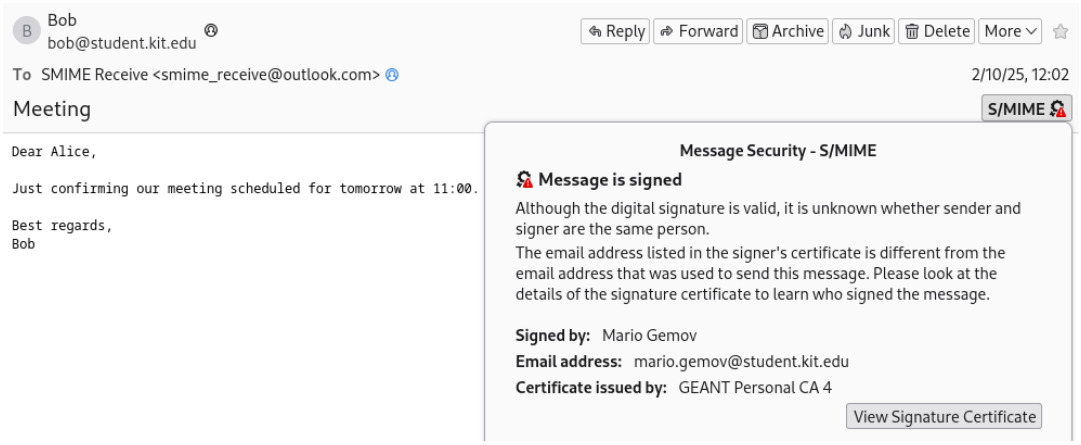

Figure 5: Thunderbird showing a warning when opening an email signed by someone other than the sender.

A.4 Failed Detection Example

Figure 6 shows how KMail fails to detect the same issue.

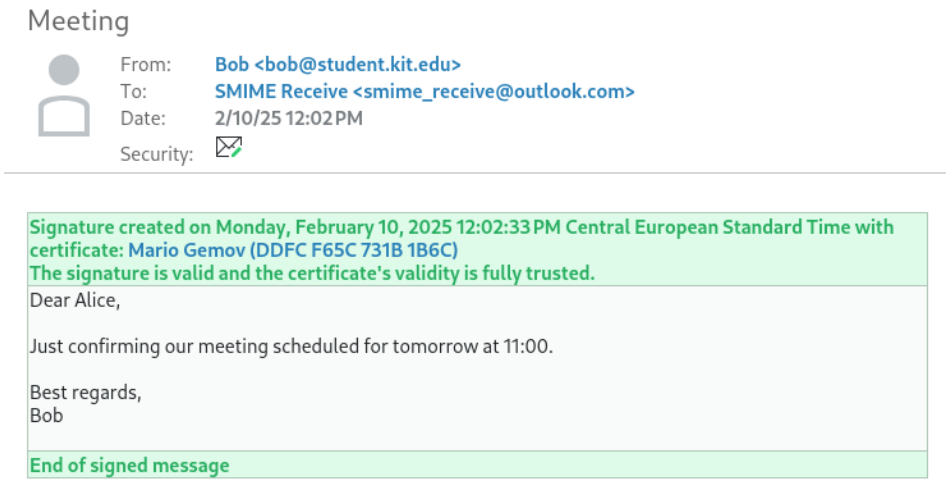

Figure 6: KMail showing that the certificate and signature are valid despite the mismatch in email addresses.