Assignment 5.1: Public-Key Cryptography

Joshua A. Wortz

University of San Diego

CSOL-510

Prof. Hoffman

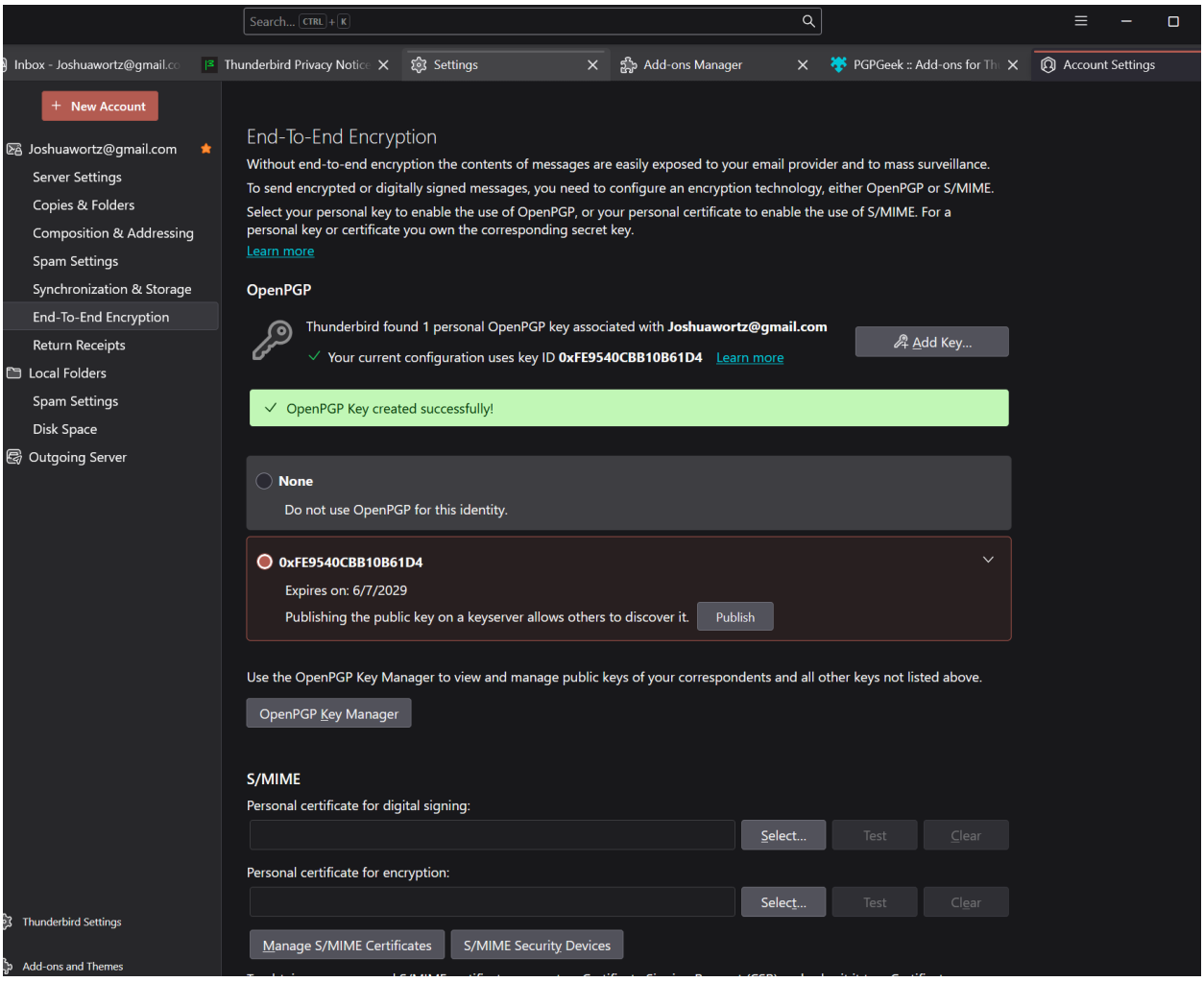
06/08/2026

Technical Foundation Implementation

To establish a functional public-key cryptography environment, Thunderbird was installed and configured to use OpenPGP. Using Thunderbird's built-in End-To-End Encryption settings, a personal OpenPGP key pair was successfully generated for the associated email identity.

Figure 1

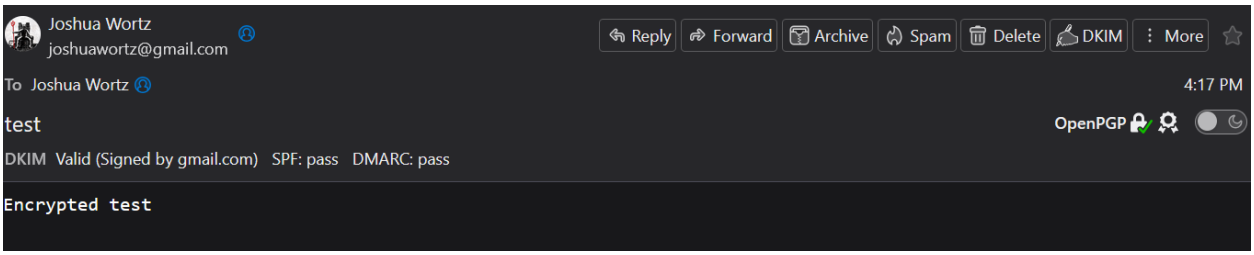
Showing OpenPGP Key successfully created with key ID 0xFE9540CBB10B61D4



To verify the successful configuration of encryption and digital signing workflows, a test email was sent to the same address. Thunderbird successfully signed and encrypted the outgoing message using the newly generated keys. Upon receipt, the email client successfully verified the signature and decrypted the message body, indicated by the OpenPGP lock and ribbon icons with a green checkmark.

Figure 2

Showing the received "test" email with the OpenPGP lock/check icons and decrypted text



Algorithms

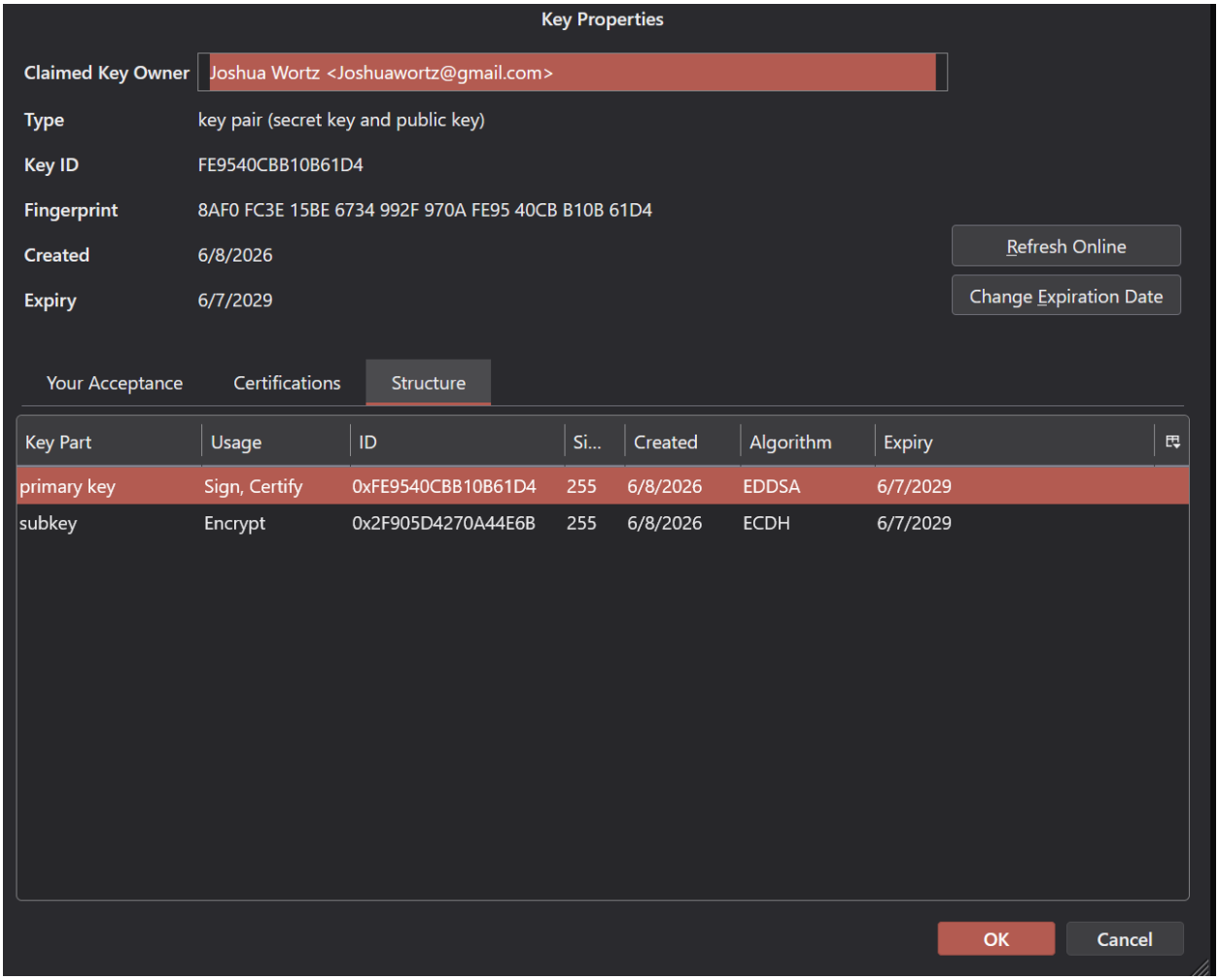
The OpenPGP implementation utilized in this setup relies on Elliptic Curve Cryptography (ECC), a modern public-key cryptography family (Mammeri, 2024). According to the Key Properties configured in Thunderbird, the key pair utilizes the following specific algorithms:

Signing and Certification (Primary Key): The primary key uses the Edwards-curve Digital Signature Algorithm (EdDSA) with a 255-bit key size. EdDSA utilizes Edwards curves (such as Ed25519) and is known for being highly secure, fast, and resilient against certain side-channel attacks (Mammeri, 2024).

Encryption (Subkey): The subkey uses Elliptic Curve Diffie-Hellman (ECDH) with a 255-bit key. ECDH is a key agreement protocol that securely computes a shared secret using scalar multiplication, which is then used to encrypt the message contents (Mammeri, 2024).

Figure 3

Showing the Key Properties, EDDSA, and ECDH algorithms



Alignment with NIST Guidelines

The ECC algorithms and 255-bit key sizes used in this OpenPGP implementation meet current security standards and align with National Institute of Standards and Technology (NIST)

cryptographic guidelines (NIST, 2026). ECC provides a robust level of security with significantly smaller key sizes than traditional integer-factorization cryptography, such as RSA (Mammeri, 2024).

The security of ECC relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which makes it computationally infeasible for attackers to find the scalar multiplier (Mammeri, 2024). Because of this exponential increase in security relative to key size, a 256-bit ECC key offers a security strength comparable to a 3072-bit RSA key (Mammeri, 2024). Therefore, utilizing 255-bit EdDSA and ECDH keys comfortably satisfies high-level security recommendations for confidentiality and authentication.

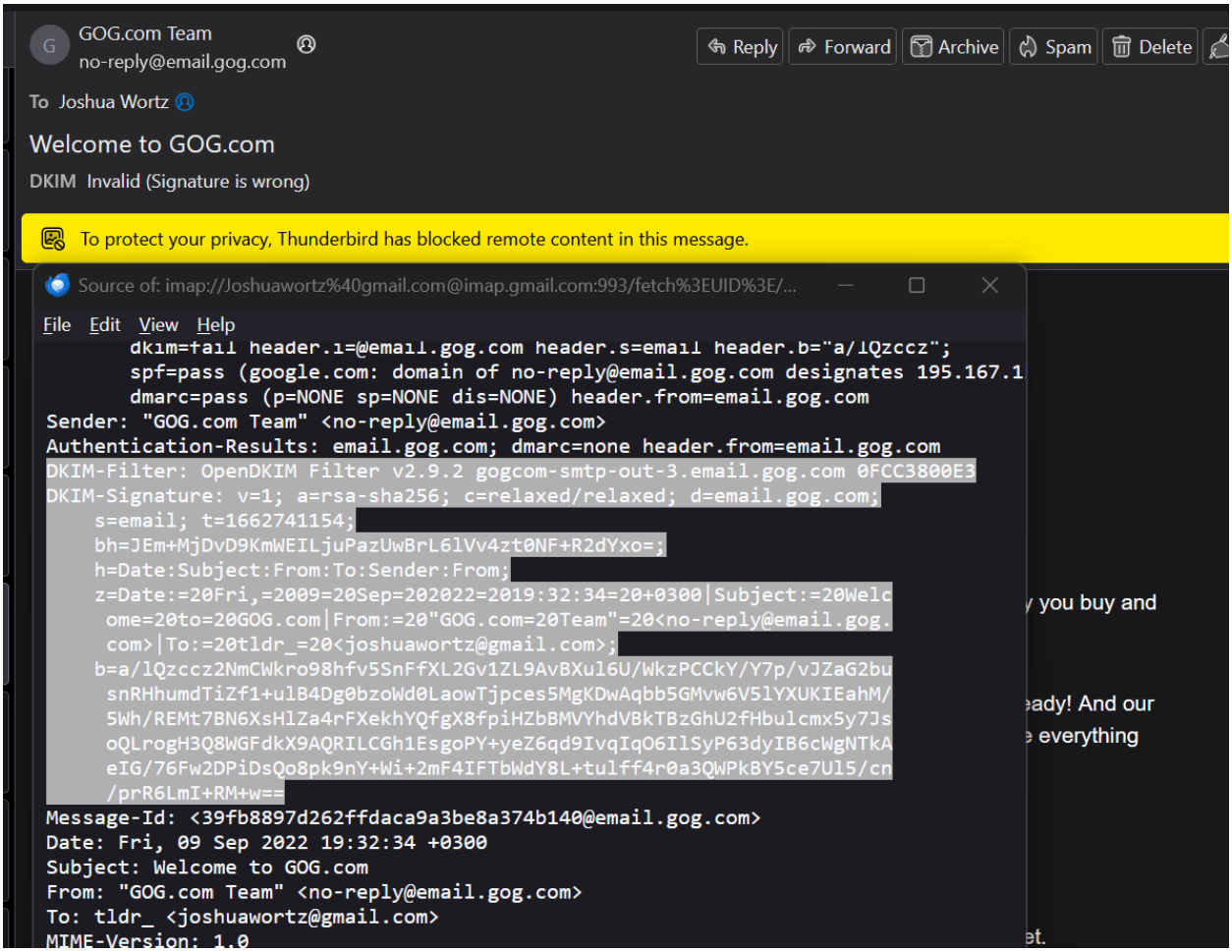
DKIM Analysis: Trusted vs. Untrusted Emails

DomainKeys Identified Mail (DKIM) is an authentication method that allows domains to cryptographically sign emails, enabling recipients to verify which domain is responsible for the message. Using a DKIM verifier in Thunderbird, incoming emails were analyzed to distinguish between trusted and untrusted signatures.

Untrusted Email Analysis: An automated marketing email claiming to be from GOG.com failed verification. The DKIM add-on flagged the message as "DKIM Invalid (Signature is wrong)" and the raw headers explicitly returned a dkim=fail result. The headers revealed an attempt to use the RSA-SHA256 algorithm, but the signature could not be validated against the domain's public records. Because GOG.com is a domain known to sign its legitimate emails, this invalid signature strongly indicates an untrusted, spoofed, or tampered message.

Figure 4

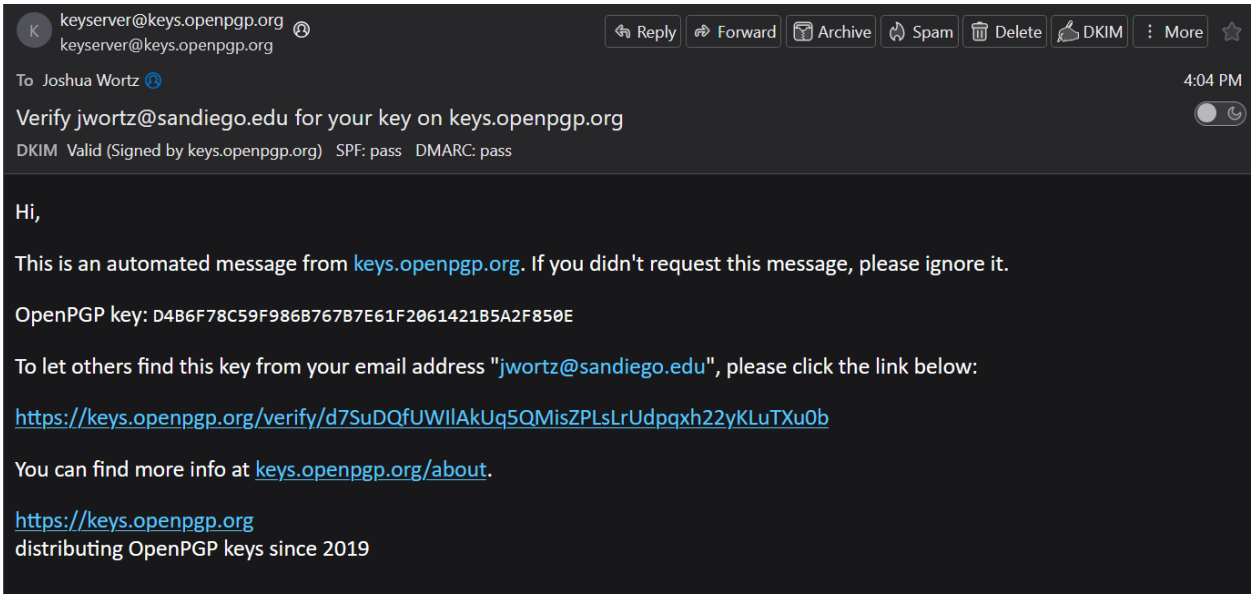
Showing the failed GOG.com DKIM check and email headers



Trusted Email Analysis: In contrast, several trusted messages were analyzed and passed the DKIM verification successfully. For example, automated verification emails received from keys.openpgp.org displayed "DKIM Valid (Signed by keys.openpgp.org)", and the self-sent test email showed "DKIM Valid (Signed by gmail.com)". These successful validations confirm that the respective domains took responsibility for the emails and that the message contents remained unaltered in transit.

Figure 5

Showing the keys.openpgp.org email with a valid DKIM signature



Analysis and Solution: Key Authenticity

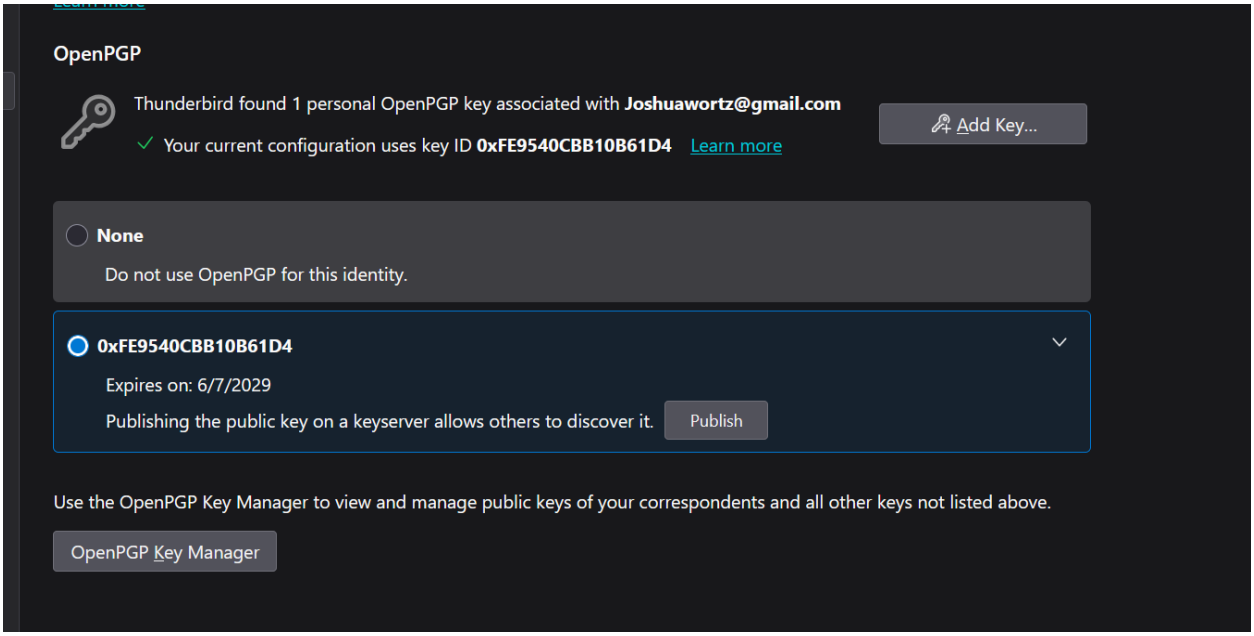
One of the primary challenges in public-key cryptosystems is verifying the authenticity of public keys. Because public keys are inherently designed for open sharing, they introduce vulnerabilities, including man-in-the-middle attacks and identity theft (Mammeri, 2024). If a user receives a public key directly from an unverified source, an attacker could easily substitute their own key, allowing them to intercept and decrypt sensitive communications.

In practice, this identity theft threat is mitigated by establishing a root of trust that binds the cryptographic key to the user's actual identity. In a traditional Public Key Infrastructure (PKI), trusted third parties known as Certificate Authorities (CAs) verify identities and issue digital certificates (Mammeri, 2024).

For OpenPGP, a practical solution to this challenge is utilizing public key servers that enforce identity verification. During this exercise, the newly generated public key was published to the public directory keys.openpgp.org.

Figure 6

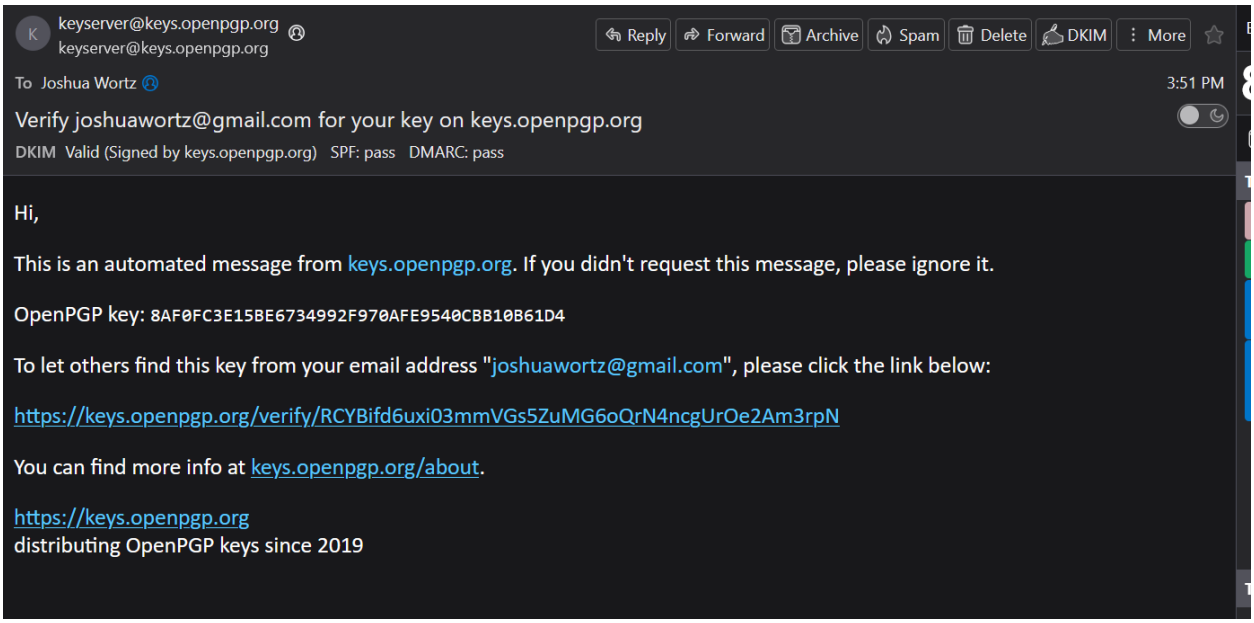
Showing the "Publish" button in the Thunderbird Key Manager



Rather than blindly accepting the upload, the server automatically dispatched an encrypted challenge to the claimed email addresses.

Figure 7

Showing the keys.openpgp.org automated verification email



The key becomes publicly searchable and trusted only after the user clicks the unique link in the email, thereby proving they control the inbox associated with the key's metadata. This mechanism binds the public key to a verified identity, effectively preventing attackers from effortlessly spoofing or distributing false keys.

References

Mammeri, Z. (2024). Cryptography: Algorithms, protocols, and standards for Computer Security (1st ed.). John Wiley & Sons.

National Institute of Standards and Technology. (2026, February 2). Cryptographic Standards and Guidelines. Information Technology Laboratory, Computer Security Resource Center. https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines