Testing & Evaluation Sheet	
Cryptomator	
1. Tool Overview	
Name:	Cryptomator
Category:	Encryption
Purpose:	Designed to provide client-side encryption for files stored in the cloud
Date Tested	4/22/25
Status:	Deployed ☑ Operational - Actively running/maintained ☐ In Testing - Currently being evaluated or piloted ☐ Inactive/Deprecated - No longer maintained or functional
Deployment Architecture:	☑ A standalone software - Runs entirely locally (e.g., runs on computer and doesn't depend on external server) ☐ A locally hosted service with separate server and client component - Run both backend/frontend yourself (e.g., backend could be on a local network, or self-hosted on cloud) ☐ A service with a local client that's hosted by a third party - You install a client on your device, but it connects to and depends on a remote server (e.g., Signal: install app (client), but Signal's servers handle message relaying, etc.) ☐ A service that is hosted by a third party but can also be self-hosted
Version:	V1.15.3
2. Installation & Setup	
OS Compatibility	Windows, macOS, Linux, Android, iOS
Installation Manual:	Yes
Installation Steps:	• Download from https://cryptomator.org/ • Standard installation (no advanced config needed for basic use)

	● Optional: install via package manager (brew install cryptomator, apt, etc.)
Mention if command-line setup or special configurations are needed	No command Line setup
Common Installation Issues & Fixes:	● Installation blocked by Microsoft Defender SmartScreen or macOS Gatekeeper ○ Click *"More Info" > "Run Anyway* ○ Alternatively, adjust Defender settings under Windows Security > App & browser control ● Missing Java Runtime or Incompatible JDK ○ Ensure Java 17+ or JDK 23 is installed and correctly configured ● Cryptomator opens with no visible interface or unresponsive menu ○ Reset application configuration ○ Delete or rename user config file (e.g. Cryptomator.cfg ○ Windows: %LocalAppData%\Cryptomator\Cryptomator.cfg ○ macOS: ~/Library/Application Support/Cryptomator/
User Documentation:	Yes
Required Technical Knowledge	Beginner

3. Testing & Evaluation

Category	Details	Score
Operational Functionality:	**Functionality** ● Test Steps: Verify the tool's core features by using all major functions, tracking any failures or bugs. ☐ The tool is mostly non-functional with many broken features and bugs. ☐ Several broken features or bugs ☐ Minor bugs or issues ☐ Mostly functional with few bugs or no bugs ☑ Fully functional with no bugs	5

	Internet Dependence: ● Allowed full access to previously synced vaults and file decryption. ● Encryption and decryption are fully performed locally. **Localization & Language Support** ● Available in over 50 languages including Chinese (Simplified and Traditional), Japanese, and Korean. ● There is an active open-source community contributing. ● Community feedback is integrated into releases regularly, with good versioning support for language packs. **Mobile Accessibility** ● Available as a mobile app for both Android and iOS.	
Usability for Non-Technical Users	**Ease of Installation & Deployment** ● One click download with password setup after ● There is video and image tutorials for download steps ● 3 minute installation. **User Onboarding Experience** ● Includes in-app tool-tips and a brief tutorial when first launching the application. ● New users are guided through vault creation and file encryption **Technical Experience Level Required** ● Users with no programming can install and start using the tool ● The interface is visual and menu-driven	4.3
Security & Privacy Strength	**Encryption Standards** ● E2EE Used ● AES-GCM (256-bit): Used for both file content and file header encryption. ● ECDH-ES (Elliptic Curve Diffie-Hellman Ephemeral Static): Used for key exchanges, specifically wrapping the user key and device-specific secrets in JSON Web Encryption (JWE) format. ● PBES2-HS256+A128KW: Used for Account Key derivation from user passwords (password-based key wrapping).	4.2

	Known Strength resilience ● Does **not rely on centralized services**, allowing private/self-hosted deployments ● Encrypts metadata, offering a level of **plausible deniability**. ● If server traffic is monitored, *metadata leakage* ● No built-in circumvention tools ● Any known weaknesses or risks? **Comparison with Known Standards** ● Strong Alignment with Industry Standards: ● Uses NIST-approved algorithms (AES-GCM, ECDH) ● Key rotation and forward secrecy mechanisms mirror best practices **Data Minimization** ● Zero-Knowledge Architecture **Privacy Policy Accessibility and Clarity** ● Only collects essential personal data ● Users can request access, correction, or deletion of their data. ● No Third-Party Advertising Trackers ● Retains minimal necessary data for legal and fulfillment purposes	
Maintenance/Sustainability	**Community support** ● Users can post questions and receive help on the official community forum. ● GitHub issues: Public bug reports, feature requests, and developer responses ● Clear, well-maintained guides and FAQs available ● Paid users (e.g., via app store purchases) can access prioritized support. **Development active status** ● Updated weekly/monthly ● Very responsive development team **Funding and Sponsorship** ● Skymatic GmbH (core development company). ● User contributions: Direct donations, GitHub Sponsors, and app store purchases. ● No major NGO or government funders ● 390k+ downloads and daily users	4.6

Performance / Effectiveness & Reliability	**Testing Environment Setup:** ● **Device:** HP Envy x360 ○ 13th Gen Intel(R) i7 processor ○ 16 GB RAM ● **Windows 11** ● **Network:** 4G Network **User Experience Observations** ● Very smooth and loads quickly **Speed & Responsiveness:** ● Small load time with no lag during use **Resource Usage:** ● 1% CPU usage ● 138.0 MB of memory **Network Performance:** ● Not applicable—tool is can be used offline for encrypting files **Reliability** ● Cryptomator has been publicly audited by Cure53, a respected cybersecurity firm. In July 2017, Cure53 conducted a white-box cryptographic audit of Cryptomator's core libraries ● Cure53's overall conclusion stated Cryptomator had a very small attack surface and no threats to long-term integrity	4
Deployment Considerations:	**Open Source & Transparency:** ● Fully open source **Cloud vs. Local Deployment:** ● Runs locally on desktops and mobile devices ● Does not require cloud infrastructure ● For cloud storage integration, users must install their cloud provider's sync client (e.g., Dropbox, OneDrive) or connect via WebDAV. **Dependencies:** ● Uses Java and JavaFX ● Dependencies are well-documented in the GitHub repositories **Post-Deployment Maintenance** ● Desktop and mobile apps are actively maintained and easy to update	

	• Forkable with a clear modular structure, making it reasonably easy to modify **Merge/Sustainability:** • Community discussions and developer engagement are encouraged via GitHub Discussions	
4. Testing Scenarios		
• **Storing Documents in an Encrypted**	• Create a new vault using Cryptomator • Stored a variety of file types (PDFs, Word docs, images) • files are accessible only when the vault is unlocked	
• **Vault Across Multiple Devices**	• Installed the Cryptomator app on my iOS device to test vault accessibility across platforms. • In order to sync vault to the iOS app, I had to connect it through Google Drive, as the mobile app requires a cloud-based vault • After syncing my Google Drive account on the iOS Cryptomator app, I was able to view the vault I created on my laptop. • He vault is interoperable between desktop and mobile platforms, although it does rely on cloud sync for access	
5. Insights & Recommendations		
Key Findings	**Strengths:** • Local-First Encryption: Client-side encryption means your data is encrypted before it reaches the cloud. • Cross-Platform: Available on Windows, macOS, Linux, Android, and iOS. • Zero-Knowledge: The app never has access to your passwords • Allows vault recovery if password is lost **Weaknesses:** • No built-in syncing; relies entirely on third-party cloud sync clients • Paid Features for Mobile: While the desktop versions are free, iOS and Android apps require a one-time purchase • Vaults need to be manually created in cloud storage	
Suggested Improvements	Improve defaults by using stronger password entropy	
Alternative Tools:	Veracrypt	

License	GPLv3
Cost/Resource Implications	**Total Cost of Ownership:** ● Desktop Platforms (Windows, macOS, Linux): ○ Free to use under **GPLv3 license** ● Mobile Platforms: ○ Android: Paid app (€19.99 incl. VAT) via Play Store or ProxyStore ○ iOS: Free Read-Only Mode ○ €19.99 for full access (lifetime license, incl. VAT)
Why is this useful to civil societies in authoritarian environments?	● Cryptomator encrypts files on the user's device so even if the cloud infrastructure is monitored or compromised by authoritarian governments like China, the actual file contents remain inaccessible. ● Cryptomator doesn't require registration or cloud-based logins, which means it doesn't tie user activity to identifiable accounts ● It works on Windows, macOS, Linux, Android, and iOS, enabling CSOs to secure their files across diverse devices and platforms ● Since all encryption and decryption occurs locally, authoritarian regimes cannot intercept data in transit or rely on backdoors in cloud storage ● Because Cryptomator doesn't rely on a centralized authentication server or proprietary network, it's harder for governments to block or disrupt its use