

Dokumentation WhistleDrop – Sichere Whistleblower-Plattform über das Tor-Netzwerk

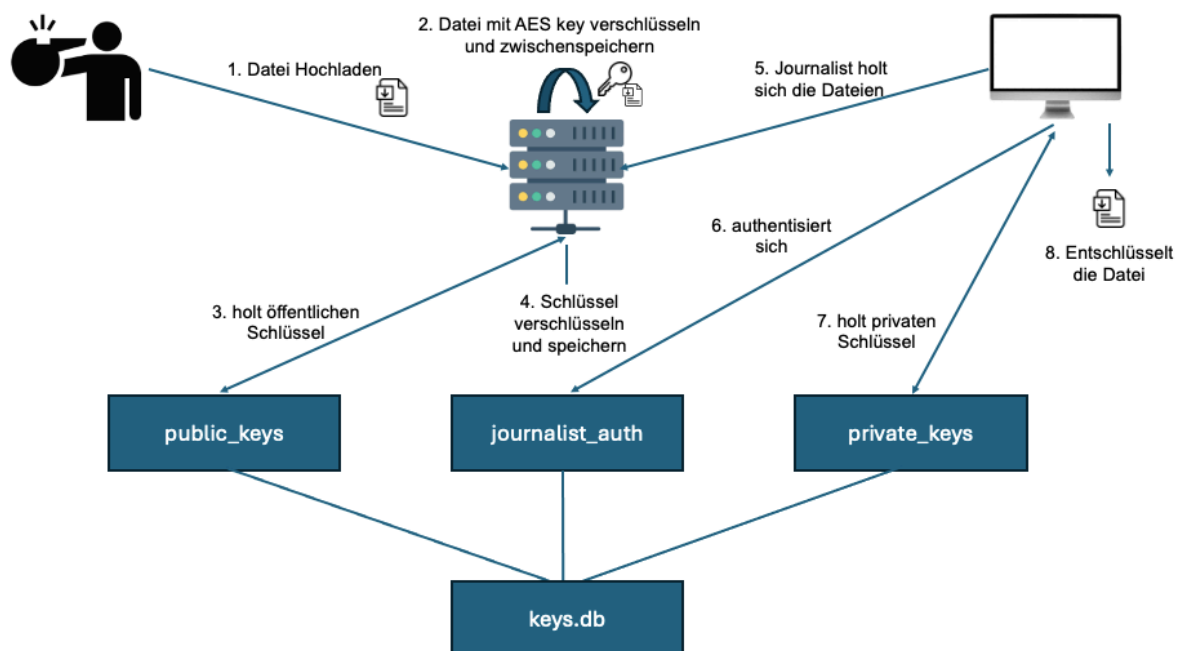
Dieses Projekt wurde gemeinsam von Kevin Wildprett(wike1017) und Tim Ruhland(ruti1016) bearbeitet.

Eine Erklärung des Whistleblowing-Prozesses mit WhistleDrop.

WhistleDrop ermöglicht eine sichere und anonyme Kommunikation zwischen einem Whistleblower und einem Journalisten über einen Tor Hidden Service. Der Whistleblower lädt über ein Webinterface eine Datei (z.B. ein PDF) auf den WhistleDrop-Server hoch. Dieser verschlüsselt die Datei sofort nach dem Hochladen mit einem zufällig generierten AES-Schlüssel (256 Bit). Der AES-Schlüssel wird anschließend mit einem einmal verwendbaren öffentlichen RSA-Schlüssel aus einer Datenbank auf dem Server verschlüsselt.

Die verschlüsselte Datei und der verschlüsselte AES-Schlüssel werden auf dem Server gespeichert und stehen dem Journalisten zum Download zur Verfügung. Zur Entschlüsselung verwendet der Journalist seinen privaten RSA-Schlüssel, der ebenfalls auf der Datenbank gespeichert und durch eine Passwortauthentifizierung geschützt ist.

Eine grafische Darstellung der Systemarchitektur. Diese soll die Interaktionen der drei Entitäten (Tabelle 1.1) und die kryptographischen Prozesse zeigen.



Schlüsselmanagement

- Jeder Journalist besitzt ein oder mehrere RSA-Schlüsselpaar(e)
- Nur die öffentlichen Schlüssel werden auf dem Server gespeichert
- Nach Verwendung wird ein öffentlicher Schlüssel automatisch gelöscht, um eine Mehrfachverwendung zu verhindern

- Die privaten Schlüssel befinden sich ausschließlich beim Journalisten
- Zugriff auf den privaten Schlüssel ist passwortgeschützt und wird aus einer lokalen SQLite-Datenbank geladen

Quellcode

Der vollständige Quellcode befindet sich im Anhang

Anforderungen aus Sicht eines Whistleblowers

Ein zentrales Ziel von WhistleDrop ist der Schutz der Identität von Hinweisgebern (Whistleblowern). Aus der Perspektive eines Whistleblowers ergeben sich mehrere zentrale Anforderungen an eine digitale Plattform für anonymes Whistleblowing. Die Erwartung absoluter Anonymität steht dabei an erster Stelle, insbesondere durch die konsequente Nutzung des Tor-Netzwerks, das die Rückverfolgung von IP-Adressen unterbindet. Des Weiteren ist es von entscheidender Bedeutung, dass unverschlüsselte Dateien zu keinem Zeitpunkt auf dem Server gespeichert oder zwischendurch abgelegt werden. WhistleDrop erfüllt diese Anforderung, indem Dateien direkt nach dem Upload mit einem zufällig generierten symmetrischen AES-Schlüssel verschlüsselt und anschließend aus dem temporären Speicher gelöscht werden.

Ein weiterer wichtiger Punkt ist die einmalige Verwendung von Schlüsseln: Öffentliche RSA-Schlüssel werden nach der Benutzung automatisch aus der Datenbank entfernt. Eine nachträgliche Wiederverwendung ist somit ausgeschlossen, was die Sicherheit zusätzlich erhöht. Darüber hinaus werden keinerlei personenbezogene Daten gespeichert oder protokolliert, sodass auch im Falle eines kompromittierten Servers keine Rückschlüsse auf die Identität des Whistleblowers möglich sind. Die Benutzeroberfläche ist bewusst auf das Wesentliche reduziert, um technische Hindernisse zu vermeiden und eine einfache und intuitive Nutzung zu gewährleisten.

Überlegen Sie sich ein Szenario, wie ein Angreifer WhistleDrop attackieren könnte und schlagen Sie eine Gegenmaßnahme vor.

Ein Angriffsszenario besteht darin, dass ein Angreifer physischen oder administrativen Zugriff auf den Server erlangt und versucht, sensible Daten abzugreifen. Ziel könnte beispielsweise sein, unverschlüsselte Inhalte oder kryptographische Schlüssel auszulesen. WhistleDrop begegnet diesem Risiko mit mehreren Schutzmaßnahmen. Zum einen wird die vom Whistleblower hochgeladene Datei unmittelbar nach dem Upload lokal mit einem 256-Bit-AES-Schlüssel verschlüsselt, der selbst nie im Klartext auf dem Server gespeichert wird. Stattdessen wird dieser AES-Schlüssel mit einem öffentlichen RSA-Schlüssel verschlüsselt, welcher aus einer kontrollierten Datenbank stammt.

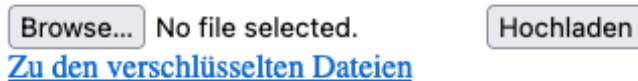
Ein weiterer Schutzmechanismus ist die automatische Löschung des öffentlichen Schlüssels nach seiner Verwendung, sodass dieser nicht mehrfach verwendet oder nachträglich analysiert werden kann. Die verschlüsselte Datei sowie der verschlüsselte AES-Schlüssel werden getrennt gespeichert, wodurch ein zusätzlicher Schutz gegen das Auslesen sensibler Informationen entsteht. Sollte ein Angreifer dennoch Zugriff auf die verschlüsselten Dateien erhalten, wären diese ohne den passenden privaten Schlüssel des Journalisten unbrauchbar.

Der Journalist entschlüsselt die Dateien zudem lokal, was eine potentielle Man-In-The-Middle Attacke verhindert.

Demonstration

1. Whistleblower verbindet sich mit der Website folgende Seite ist zu sehen:

Datei für Verschlüsselung hochladen



Browse... No file selected. Hochladen

[Zu den verschlüsselten Dateien](#)

2. Der Whistleblower drückt nun auf „Browse“ ein PoP-Up Fenster öffnet sich und es lässt sich eine Datei auswählen:

Datei für Verschlüsselung hochladen



Browse... geheimnis.pdf Hochladen

[Zu den verschlüsselten Dateien](#)

3. Nach dem nun auf Hochladen gedrückt wird erscheint folgende Website und die Datei wird verschlüsselt auf dem Server gespeichert:

Datei erfolgreich hochgeladen und verschlüsselt.

[Zurück zur Upload-Seite](#)

4. Nun kann ein Journalist sich ebenfalls auf der Website einwählen und auf den „Zu den verschlüsselten Dateien“ Link drücken. In der Abbildung sehen wir unsere „geheimnis.pdf“ Datei und zu demonstrationszwecken noch eine 2. Bereits hochgeladene Datei. Nun kann sich der Journalist die verschlüsselten Dateien sichern.

Verschlüsselte Dateien und Schlüssel

- geheimnis.pdf [geheimnis.pdf.enc](#) | [geheimnis.pdf.key](#)
- Noch_geheimeres_Geheimnis.docx [Noch_geheimeres_Geheimnis.docx.enc](#) | [Noch_geheimeres_Geheimnis.docx.key](#)

[Zurück zur Upload-Seite](#)

5. Der Journalist kann das Entschlüsselungsskript ausführen. Und wird zunächst nach der Datei gefragt, welche er entschlüsseln möchte:


```
Bitte gib den Basisdateinamen der Datei ein (z.B. bericht.docx):
```

6. Im nächsten Schritt wird der Journalist zur Absicherung der privaten Schlüssel nach seinem Zugang gefragt. Dieser wird mittels der Erzeugung der privaten Schlüssel für den Journalisten angelegt:

```
Journalist-ID: wike1017  
Passwort: geheim123
```

7. Nach erfolgreicher Authentifizierung wird die Entschlüsselung durchgeführt:

```
[+] Entschlüssele AES-Schlüssel...  
[+] Entschlüssele die Datei...  
[+] Speichere entschlüsselte Datei als entschluesselt_geheimnis.pdf...  
[+] Entschlüsselung abgeschlossen.  
[!] Privater Schlüssel wurde aus der Datenbank gelöscht.
```

 entschluesselt_geheimnis.pdf

8. Kumulierte Arbeitszeit

Die kumulierte Arbeitszeit der gesamten Gruppe liegt bei ca. 12h.