Encrypted Container File

Design and Implementation of a Hybrid-Encrypted Multi-Recipient File Structure

27th June 2023

Tobias J. Bauer and Andreas Aßmuth | Ostbayerische Technische Hochschule Amberg-Weiden
t.bauer@oth-aw.de a.assmuth@oth-aw.de

Tobias J. Bauer

B. Eng. Media Informatics at OTH Amberg-Weiden, Germany, 2022

Currently in Master's Degree Program Artificial Intelligence (M. Sc.), est. 2023

Interests in Infrastructure Security and the application of AI in IT Sec

Outline

Introduction and Related Work

Design of the Encrypted Container File

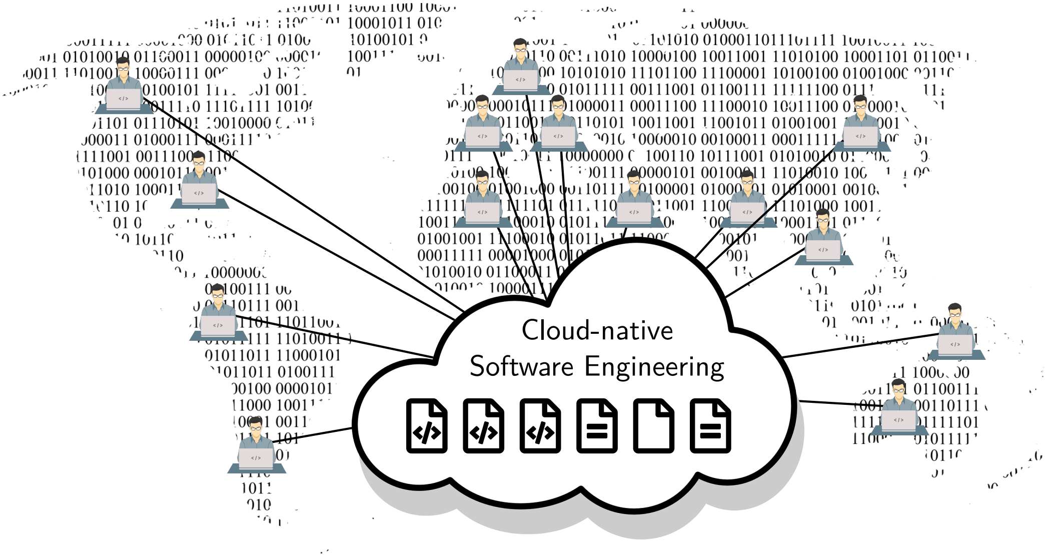
- Requirements Engineering
- File Structure
- Operations

Implementation Details

Conclusion and Future Work

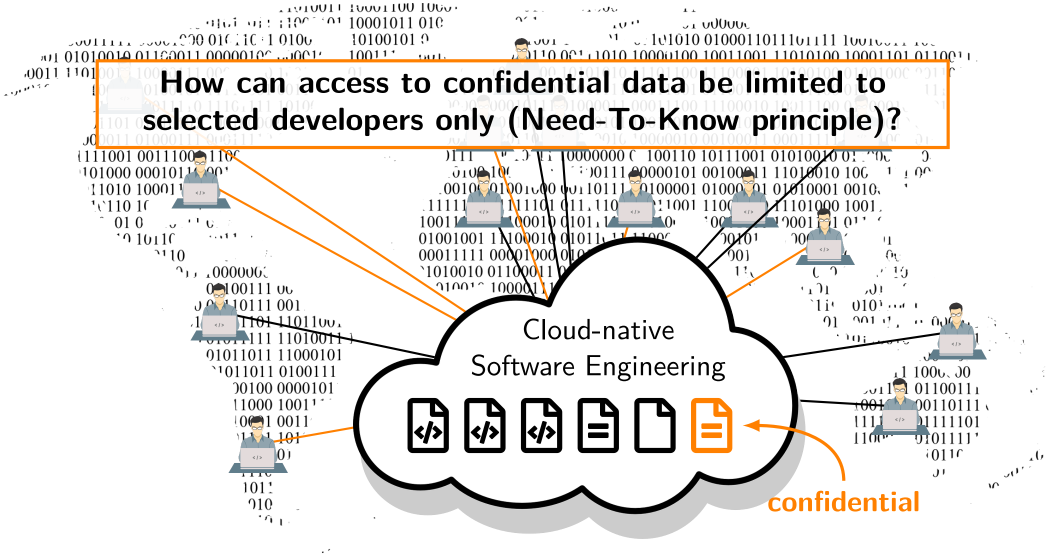
Motivation

Motivation



Motivation

How can access to confidential data be limited to selected developers only (Need-To-Know principle)?



Cloud-native
Software Engineering

confidential

Related Work

Introduction and Related Work

jak [1]

- Single command encryption and decryption (AES)
- Single key for all confidential files
- Unencrypted files on developers' computers
- Key distribution problem unsolved

Related Work

Introduction and Related Work

jak [1]

- Single command encryption and decryption (AES)
- Single key for all confidential files
- Unencrypted files on developers' computers
- Key distribution problem unsolved

git-crypt [2]

- Single command encryption and decryption (AES)
- Single key for all confidential files
- Unencrypted files on developers' computers
- GNU Privacy Guard for key distribution
- No recipient removal

Design of the Encrypted Container File

- Requirements Engineering
- File Structure
- Operations

Requirements Engineering

Design of the Encrypted Container File

Requirements

Design goals

Requirements Engineering

Design of the Encrypted Container File

Requirements

(1) Mandatory encryption of content

(2) Possibility to modify content

(3) Key distribution is no prerequisite

(4) Decryption on demand

(5) Support for multiple recipients

(6) Addition and removal of recipients

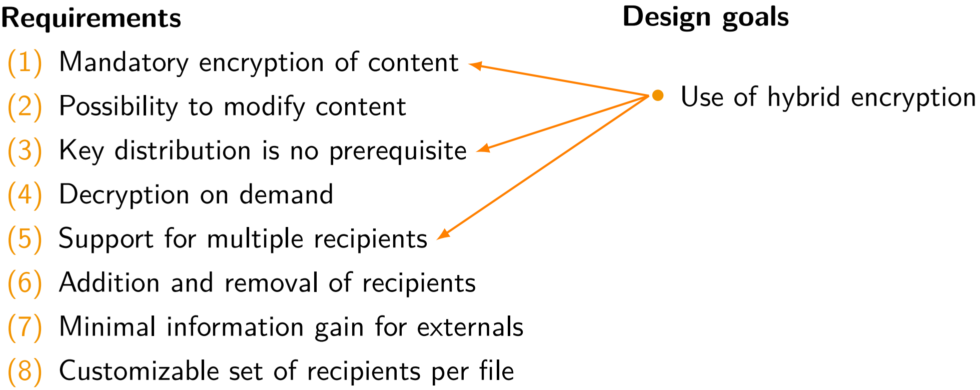
(7) Minimal information gain for externals

(8) Customizable set of recipients per file

Design goals

Requirements Engineering

Design of the Encrypted Container File



Requirements

(1) Mandatory encryption of content

(2) Possibility to modify content

(3) Key distribution is no prerequisite

(4) Decryption on demand

(5) Support for multiple recipients

(6) Addition and removal of recipients

(7) Minimal information gain for externals

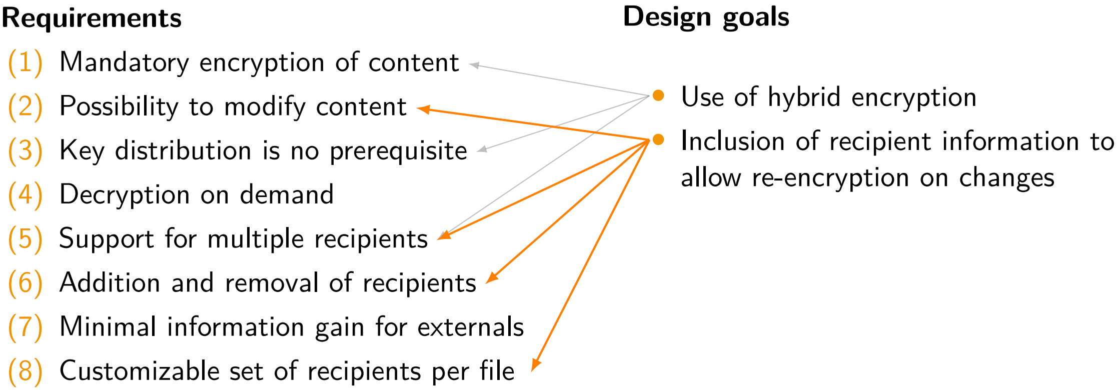
(8) Customizable set of recipients per file

Design goals

- Use of hybrid encryption

Requirements Engineering

Design of the Encrypted Container File



Requirements

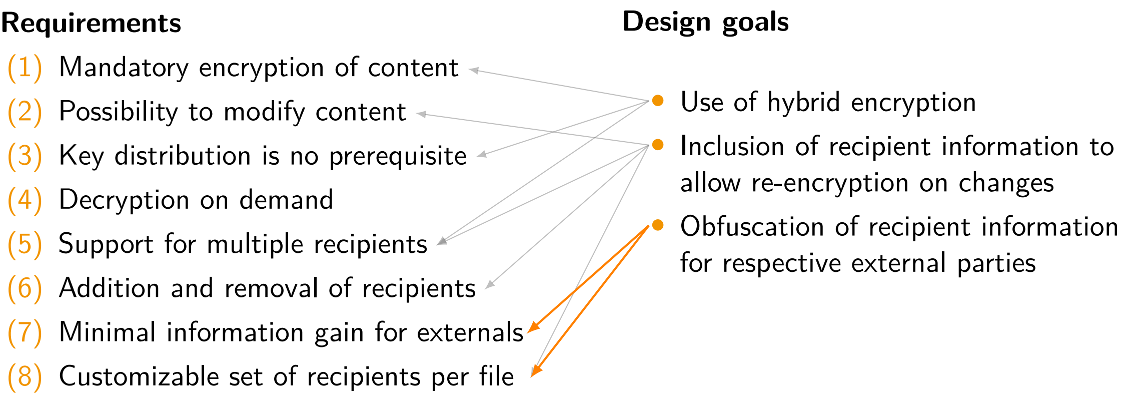
(1) Mandatory encryption of content
(2) Possibility to modify content
(3) Key distribution is no prerequisite
(4) Decryption on demand
(5) Support for multiple recipients
(6) Addition and removal of recipients
(7) Minimal information gain for externals
(8) Customizable set of recipients per file

Design goals

- Use of hybrid encryption
- Inclusion of recipient information to allow re-encryption on changes

Requirements Engineering

Design of the Encrypted Container File



Requirements

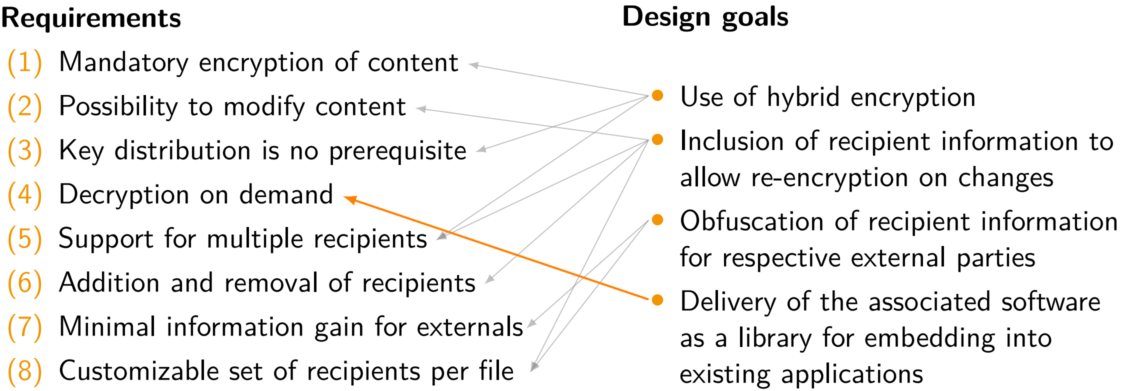
(1) Mandatory encryption of content
(2) Possibility to modify content
(3) Key distribution is no prerequisite
(4) Decryption on demand
(5) Support for multiple recipients
(6) Addition and removal of recipients
(7) Minimal information gain for externals
(8) Customizable set of recipients per file

Design goals

- Use of hybrid encryption
- Inclusion of recipient information to allow re-encryption on changes
- Obfuscation of recipient information for respective external parties

Requirements Engineering

Design of the Encrypted Container File



Requirements

(1) Mandatory encryption of content

(2) Possibility to modify content

(3) Key distribution is no prerequisite

(4) Decryption on demand

(5) Support for multiple recipients

(6) Addition and removal of recipients

(7) Minimal information gain for externals

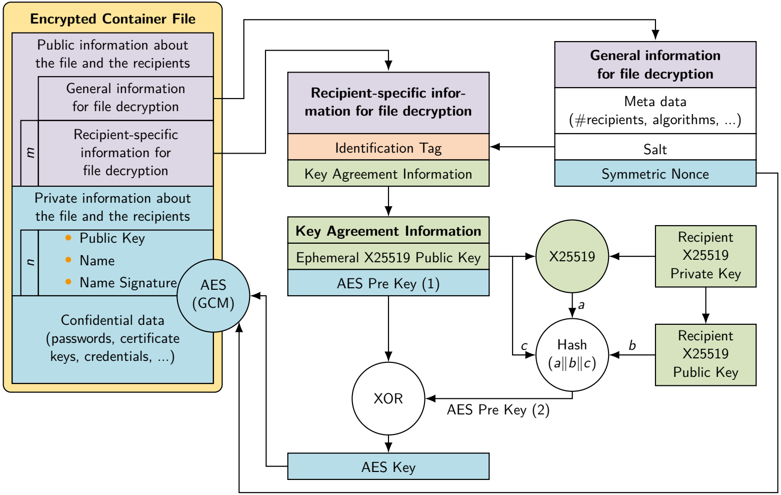
(8) Customizable set of recipients per file

Design goals

- Use of hybrid encryption
- Inclusion of recipient information to allow re-encryption on changes
- Obfuscation of recipient information for respective external parties
- Delivery of the associated software as a library for embedding into existing applications

File Structure

Design of the Encrypted Container File



Encrypted Container File

Public information about the file and the recipients

General information for file decryption

Recipient-specific information for file decryption

Private information about the file and the recipients

- Public Key
- Name
- Name Signature

Confidential data (passwords, certificate keys, credentials, ...)

Recipient-specific information for file decryption

Identification Tag

Key Agreement Information

General information for file decryption

Meta data (#recipients, algorithms, ...)

Salt

Symmetric Nonce

Key Agreement Information

Ephemeral X25519 Public Key

AES Pre Key (1)

X25519

Recipient X25519 Private Key

Hash ($a||b||c$)

Recipient X25519 Public Key

XOR

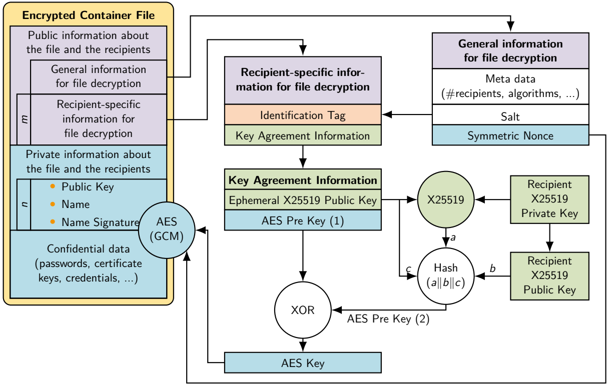
AES Pre Key (2)

AES Key

AES (GCM)

Operations

Design of the Encrypted Container File

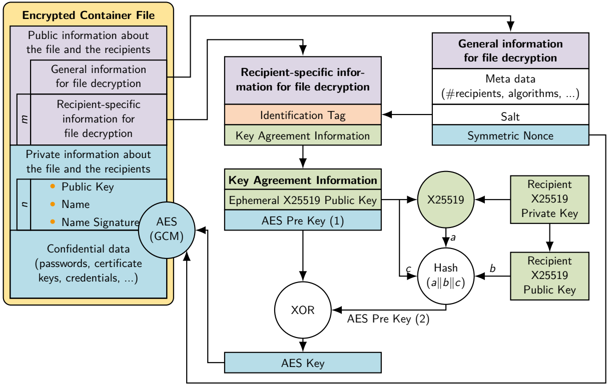

Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
b
Hash (a||b||c)
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Prerequisites for decryption

- *Alice* is recipient
- Her private X25519 key: $\mathrm{sk}_{\mathrm{A}}^{\mathrm{X}}$
- Her public X25519 key: $\mathrm{pk}_{\mathrm{A}}^{\mathrm{X}}$

Operations

Design of the Encrypted Container File

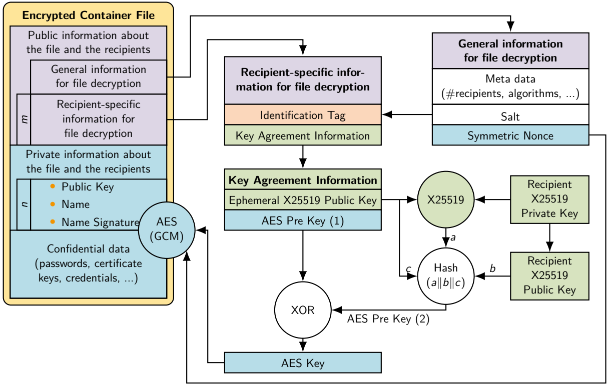

Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Prerequisites for decryption

- *Alice* is recipient
- Her private X25519 key: $\mathrm{sk}_{\mathrm{A}}^{\mathrm{X}}$
- Her public X25519 key: $\mathrm{pk}_{\mathrm{A}}^{\mathrm{X}}$
- Hash function: H
- Bit string concatenation: $a||b$
- Bitwise XOR: $a \oplus b$
- Bytewise truncation: $a[0,\ldots,n]$
- Scalar-Point-multiplication [3]: $\mathrm{X25519}(a, B)$

Operations

Design of the Encrypted Container File

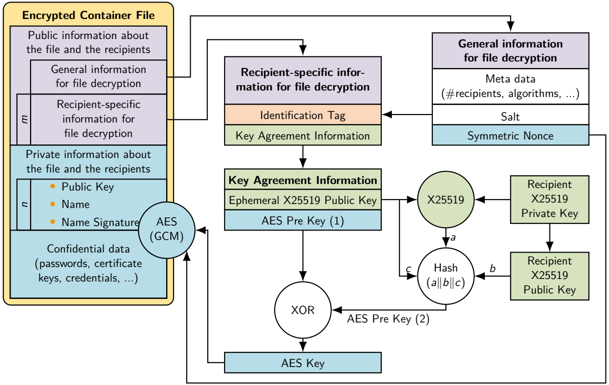

Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
b
Hash (a||b||c)
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Decryption

$$(1)\quad \text{id\_tag} = \text{H}\left(\text{pk}_{\text{A}}^{\text{Ed}} \| \text{Salt}\right)[0,\ldots,16]$$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
AES (GCM)
Confidential data (passwords, certificate keys, credentials, ...)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

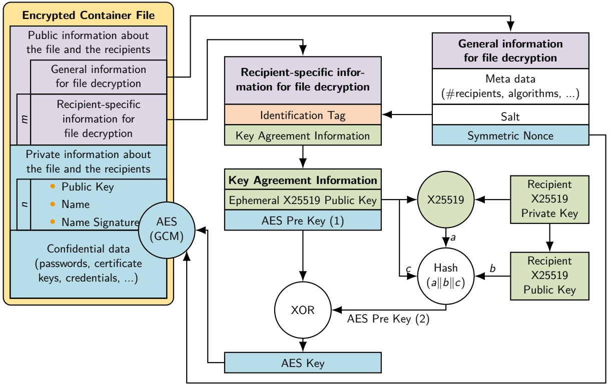
Decryption

(1) $\text{id\_tag} = H\left(\text{pk}_A^{\text{Ed}} \| \text{Salt}\right)[0,\ldots,16]$

(2) $\text{Load } \left(\text{pk}_e^{X}, \text{k}_{\text{pre1}}^{\text{AES}}\right)$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a‖b‖c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Decryption

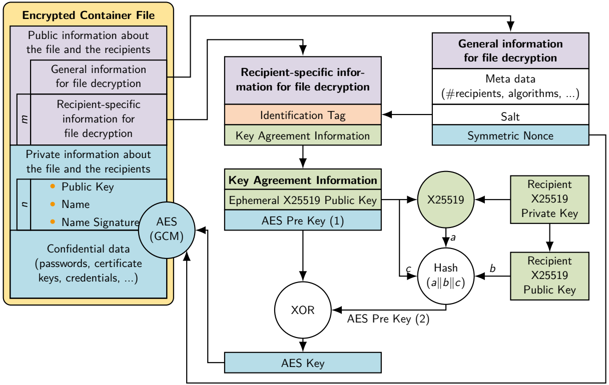
(1) $\mathrm{id\_tag} = \mathrm{H}\left(\mathrm{pk}_{\mathrm{A}}^{\mathrm{Ed}} \| \mathrm{Salt}\right)[0,\dots,16]$

(2) Load $\left(\mathrm{pk}_{\mathrm{e}}^{\mathrm{X}}, \mathrm{k}_{\mathrm{pre1}}^{\mathrm{AES}}\right)$

(3) $\mathrm{k}_{\mathrm{sh}}^{\mathrm{X}} = \mathrm{X25519}\left(\mathrm{sk}_{\mathrm{A}}^{\mathrm{X}}, \mathrm{pk}_{\mathrm{e}}^{\mathrm{X}}\right)$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
Recipient-specific information for file decryption
m
Private information about the file and the recipients
Public Key
Name
Name Signature
n
AES (GCM)
Confidential data (passwords, certificate keys, credentials, ...)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Decryption

(1) $\text{id\_tag} = H\left(\text{pk}_A^{\text{Ed}} \| \text{Salt}\right)[0,\ldots,16]$

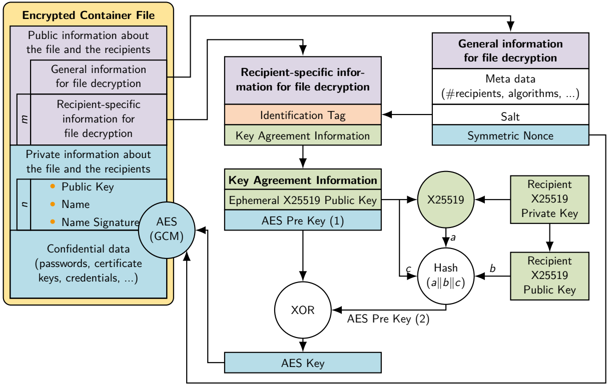
(2) Load $\left(\text{pk}_e^X, \text{k}_{\text{pre1}}^{\text{AES}}\right)$

(3) $\text{k}_{\text{sh}}^X = \text{X25519}\left(\text{sk}_A^X, \text{pk}_e^X\right)$

(4) $\text{k}_{\text{pre2}}^{\text{AES}} = H\left(\text{k}_{\text{sh}}^X \| \text{pk}_A^X \| \text{pk}_e^X\right)[0,\ldots,32]$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
Recipient-specific information for file decryption
m
Private information about the file and the recipients
Public Key
Name
Name Signature
n
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Decryption

(1) $\text{id\_tag} = H\left(\text{pk}_A^{\text{Ed}} \| \text{Salt}\right)[0,\dots,16]$

(2) Load $\left(\text{pk}_e^X, \text{k}_{\text{pre1}}^{\text{AES}}\right)$

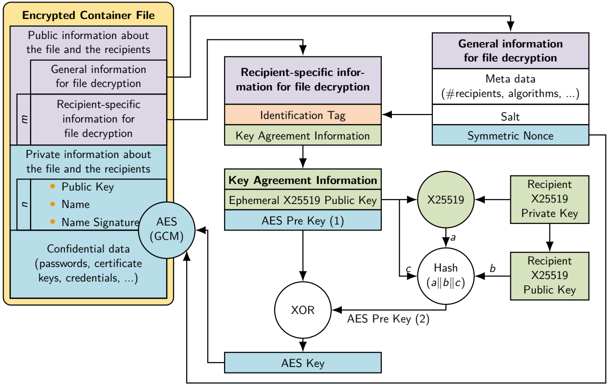
(3) $\text{k}_{\text{sh}}^X = \text{X25519}\left(\text{sk}_A^X, \text{pk}_e^X\right)$

(4) $\text{k}_{\text{pre2}}^{\text{AES}} = H\left(\text{k}_{\text{sh}}^X \| \text{pk}_A^X \| \text{pk}_e^X\right)[0,\dots,32]$

(5) $\text{k}^{\text{AES}} = \text{k}_{\text{pre1}}^{\text{AES}} \oplus \text{k}_{\text{pre2}}^{\text{AES}}$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Encryption

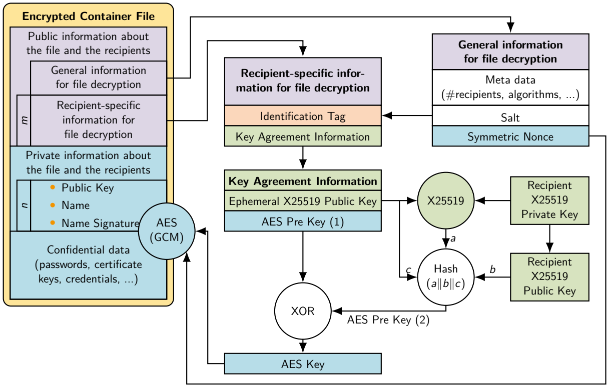
(1) Generate symmetric AES key

(2) Generate AES nonce

(3) Generate salt

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
• Public Key
• Name
• Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Encryption

(1) Generate symmetric AES key

(2) Generate AES nonce

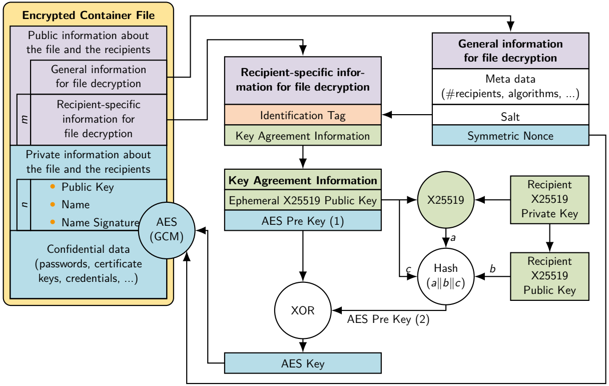
(3) Generate salt

(4) For each recipient *r*

  (a) Load pk_r^X

Operations

Design of the Encrypted Container File



Encrypted Container File

Public information about the file and the recipients

General information for file decryption

n

Recipient-specific information for file decryption

Private information about the file and the recipients

n

- Public Key
- Name
- Name Signature

Confidential data (passwords, certificate keys, credentials, ...)

AES (GCM)

Recipient-specific information for file decryption

Identification Tag

Key Agreement Information

General information for file decryption

Meta data (#recipients, algorithms, ...)

Salt

Symmetric Nonce

Key Agreement Information

Ephemeral X25519 Public Key

AES Pre Key (1)

X25519

Recipient X25519 Private Key

a

c

Hash ($a\|b\|c$)

b

Recipient X25519 Public Key

XOR

AES Pre Key (2)

AES Key

Encryption

(1) Generate symmetric AES key

(2) Generate AES nonce

(3) Generate salt

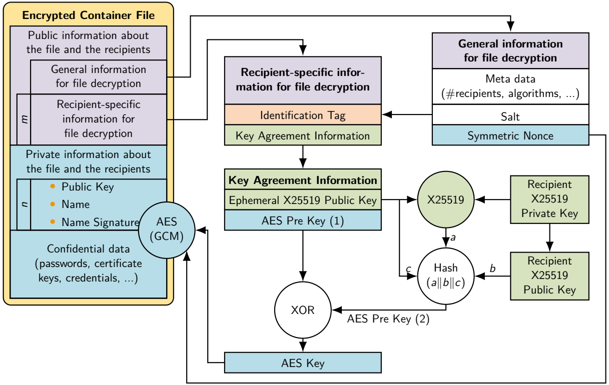
(4) For each recipient r

(a) Load pk_r^X

(b) $\left(\mathrm{sk}_e^X, \mathrm{pk}_e^X\right) \leftarrow \mathrm{Gen}^X$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Encryption

(1) Generate symmetric AES key

(2) Generate AES nonce

(3) Generate salt

(4) For each recipient *r*

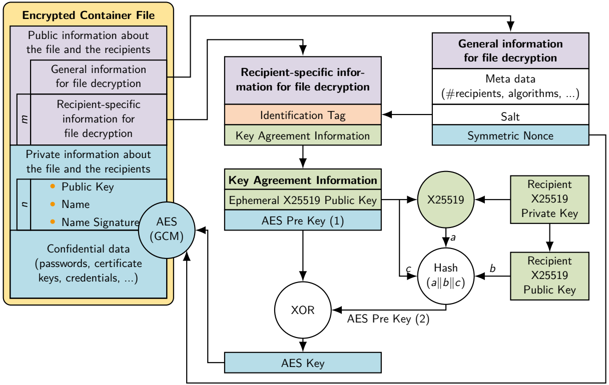
  (a) Load pk_r^X

  (b) $\left(\mathrm{sk}_e^X, \mathrm{pk}_e^X\right) \leftarrow \mathrm{Gen}^X$

  (c) Compute id_tag, $\mathrm{k}_{\mathrm{sh}}^X$, $\mathrm{k}_{\mathrm{pre2}}^{\mathrm{AES}}$

Operations

Design of the Encrypted Container File


Encrypted Container File
Public information about the file and the recipients
General information for file decryption
m
Recipient-specific information for file decryption
Private information about the file and the recipients
n
Public Key
Name
Name Signature
Confidential data (passwords, certificate keys, credentials, ...)
AES (GCM)
Recipient-specific information for file decryption
Identification Tag
Key Agreement Information
General information for file decryption
Meta data (#recipients, algorithms, ...)
Salt
Symmetric Nonce
Key Agreement Information
Ephemeral X25519 Public Key
AES Pre Key (1)
X25519
Recipient X25519 Private Key
a
c
Hash (a||b||c)
b
Recipient X25519 Public Key
XOR
AES Pre Key (2)
AES Key

Encryption

(1) Generate symmetric AES key

(2) Generate AES nonce

(3) Generate salt

(4) For each recipient r

   (a) Load pk_r^X

   (b) $\left(\mathrm{sk}_e^X, \mathrm{pk}_e^X\right) \leftarrow \mathrm{Gen}^X$

   (c) Compute id_tag, $\mathrm{k}_{\mathrm{sh}}^X$, $\mathrm{k}_{\mathrm{pre2}}^{\mathrm{AES}}$

   (d) $\mathrm{k}_{\mathrm{pre1}}^{\mathrm{AES}} = \mathrm{k}^{\mathrm{AES}} \oplus \mathrm{k}_{\mathrm{pre2}}^{\mathrm{AES}}$

Operations

Design of the Encrypted Container File

Further Operations

General procedure

(1) Decrypt Encrypted Container File
(2) Modify content and/or recipient list
(3) Encrypt Encrypted Container File

Operations

Design of the Encrypted Container File

Further Operations

General procedure

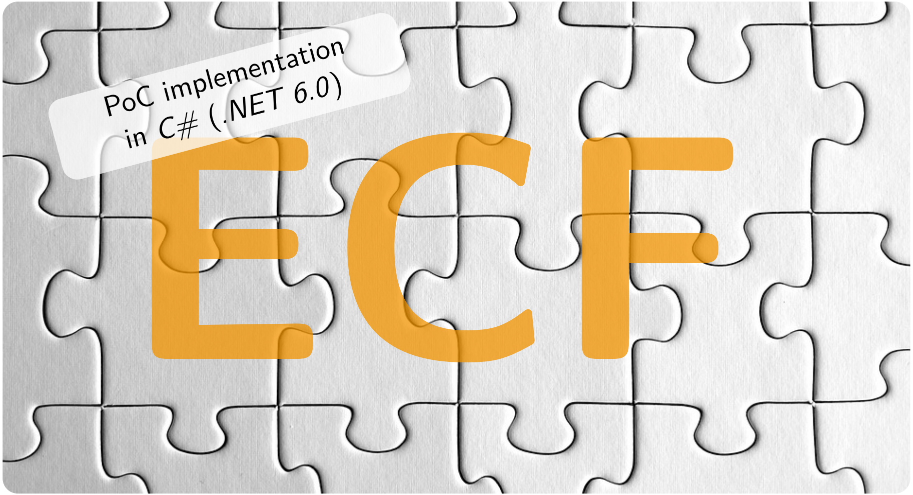
(1) Decrypt Encrypted Container File
(2) Modify content and/or recipient list
(3) Encrypt Encrypted Container File

Possible operations

- Modification of confidential data
- Addition of a new recipient
- Removal of an existing recipient

Implementation Details

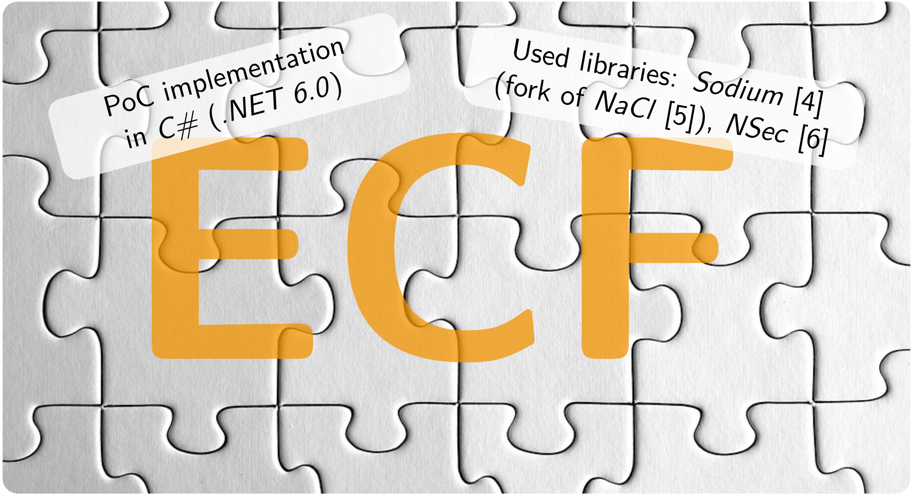
How everything is put together



PoC implementation in C# (.NET 6.0)

Implementation Details

How everything is put together

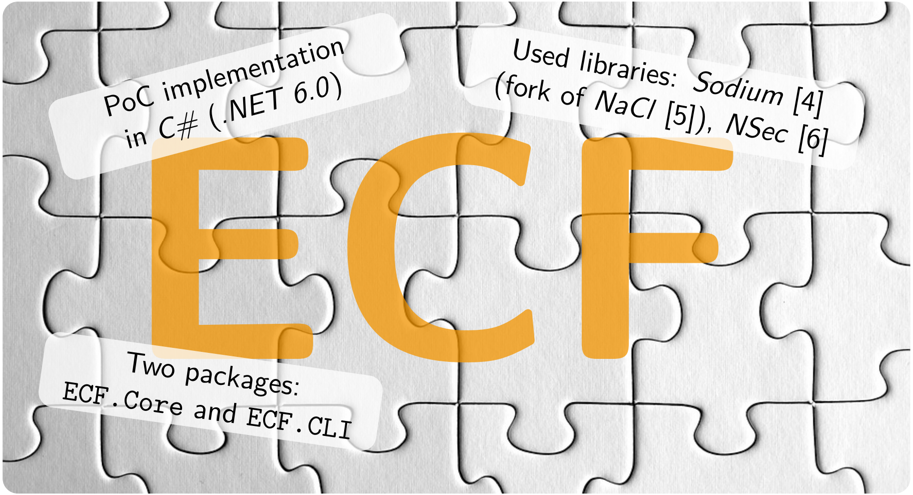


PoC implementation in C# (.NET 6.0)

Used libraries: Sodium [4] (fork of NaCl [5]), NSec [6]

Implementation Details

How everything is put together



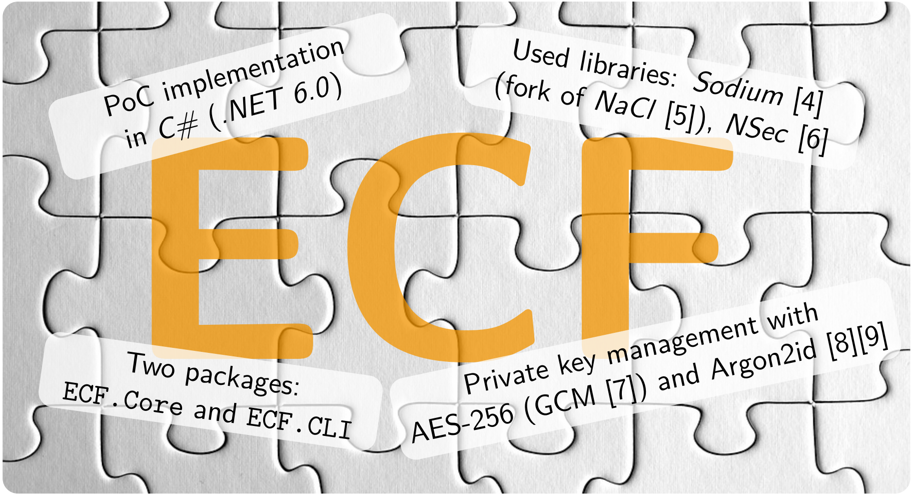
PoC implementation in C# (.NET 6.0)

Used libraries: *Sodium* [4] (fork of *NaCl* [5]), *NSec* [6]

Two packages: `ECF.Core` and `ECF.CLI`

Implementation Details

How everything is put together



PoC implementation in C# (.NET 6.0)

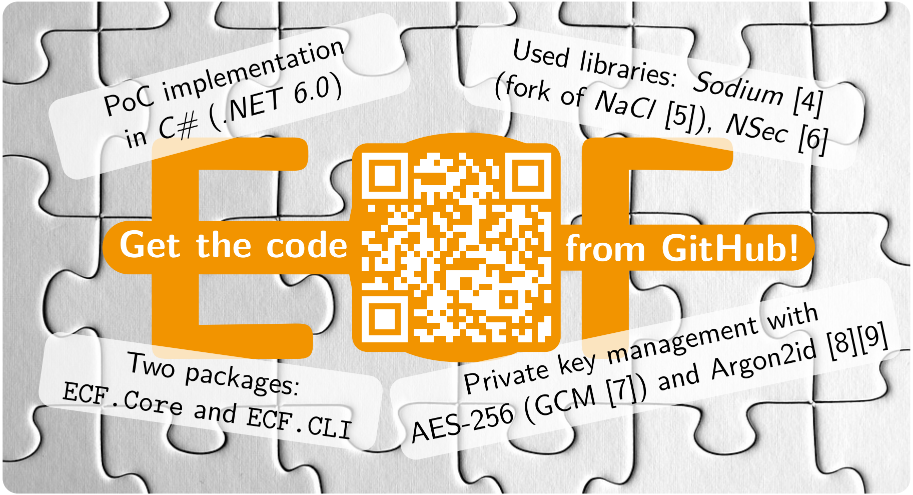
Used libraries: Sodium [4] (fork of *NaCl* [5]), *NSec* [6]

Two packages: `ECF.Core` and `ECF.CLI`

Private key management with AES-256 (GCM [7]) and Argon2id [8][9]

Implementation Details

How everything is put together



PoC implementation in C# (.NET 6.0)

Used libraries: Sodium [4] (fork of *NaCl* [5]), NSec [6]

Get the code from GitHub!

Two packages: `ECF.Core` and `ECF.CLI`

Private key management with AES-256 (GCM [7]) and Argon2id [8][9]

Conclusion and Future Work

- Proof of Concept (PoC) implementation supports two cipher suites
- Implementation of more cipher suites possible
- Full code and unit tests available:
  `https://github.com/Hirnmoder/ECF`

Wordcloud created by Ashashyou, CC BY-SA 4.0

Tobias J. Bauer, B. Eng.

Ostbayerische Technische Hochschule Amberg-Weiden
Department of Electrical Engineering, Media and Computer Science
Kaiser-Wilhelm-Ring 23, 92224 Amberg, Germany

Email: t.bauer@oth-aw.de
Web: https://www.oth-aw.de

ECF on GitHub

References I

[1] Dispel LLC, “Jak – simple git encryption,” Dispel LLC. (2017), [Online]. Available: `https://jak.readthedocs.io/en/latest/` (visited on 06/05/2023).

[2] A. Ayer, “Git-crypt – transparent file encryption in git,” (2023), [Online]. Available: `https://www.agwa.name/projects/git-crypt/` (visited on 06/05/2023).

[3] D. J. Bernstein, “Curve25519: New diffie-hellman speed records,” in *Public Key Cryptography - PKC 2006*, Springer Berlin Heidelberg, 2006, pp. 207–228.

[4] The Sodium Authors, “Introduction – libsodium,” (2022), [Online]. Available: `https://doc.libsodium.org/` (visited on 06/05/2023).

[5] D. J. Bernstein, T. Lange, and P. Schwabe, “Nacl: Networking and cryptography library,” (Mar. 15, 2016), [Online]. Available: `https://nacl.cr.yp.to` (visited on 06/05/2023).

References II

[6] K. Hartke, “Nsec – modern cryptography for .net core,” (2022), [Online]. Available: https://nsec.rocks/ (visited on 06/05/2023).

[7] D. A. McGrew and J. Viega, “The galois/counter mode of operation (GCM),” National Institute of Standards and Technology. (May 31, 2005), [Online]. Available: https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf (visited on 06/05/2023).

[8] A. Biryukov, D. Dinu, and D. Khovratovich, “Argon2: The memory-hard function for password hashing and other applications,” version 1.3. (Mar. 24, 2017), [Online]. Available: https://raw.githubusercontent.com/P-H-C/phc-winner-argon2/master/argon2-specs.pdf (visited on 06/05/2023).

[9] The Sodium Authors, “The pwhash* api,” (2022), [Online]. Available: https://doc.libsodium.org/password_hashing/default_phf (visited on 06/05/2023).