Design Phase

Thema: Cryptography API

Dokumentinformationen

Dateiname: doku-template-short.docx
Speicherdatum: 03.05.2022

Autoreninformationen

Autor: Anton Detken
E-Mail: anton@detken.ch
Tel: +41 77 474 12 33

Table of Content

Figures

No table of figures entries found.

Tables

No table of figures entries found.

Note: Table of figures & tables references are broken, therefore not showing anything.

1 Description

The API offers multiple endpoints to decrypt and encrypt data in combination with a key. There is a selection of cipher methods available.

Following ciphers are available:

- Blowfish
- Twofish
- Perhaps: AES

Following hashing algorithms are available

- BCrypt

Some ciphers might have additional specifications, such as number of rounds, that can be defined.

As this is an API, the methods are accessible via http-endpoints. Only users with an account can use the API.

2 Domain Model

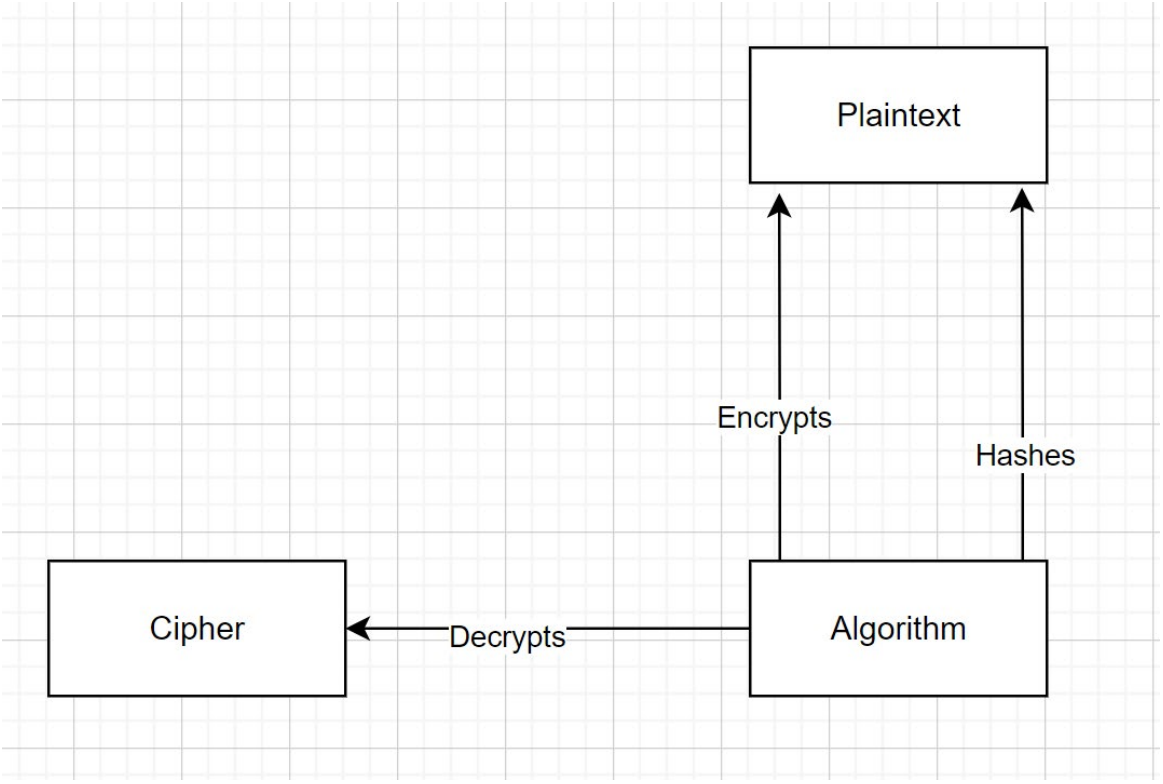

Figure 1 Domain Model

3 Use Cases

Template

Use Case #[nr]	[title]
Pre-Condition	
Main Scenario / Description of use case in detail	
Post-Condition	
Exceptions[1]	

Table 1 Use Case Template

Encryption

Use Case #1	Encryption
Pre-Condition	None, User has a plaintext / -file
Main Scenario / Description of use case in detail	Send a GET-request to the endpoint of a cipher with following parameter: - Plaintext Receive the ciphertext and generated key.
Post-Condition	User has a ciphertext and a key.
Exceptions	Loss of connection during the encryption process. Result: HTTP error code.

Table 2 Use Case 1

Encryption with custom Key

Use Case #2	Encryption
Pre-Condition	None, User has a plaintext / -file and *a custom Key*.
Main Scenario / Description of use case in detail	Send a GET-request to the endpoint of a cipher with following parameters: - Plaintext - Key Receive the ciphertext and key.
Post-Condition	User has a ciphertext and a key.
Exceptions	1: Loss of connection during the encryption process. 2: Incompatible Key size. Result: HTTP error code with message.

Table 3 Use Case 2

Decryption

Use Case #3	Decryption
Pre-Condition	None, User has a ciphertext and key

[1] What can go wrong? How will the system respond?

Main Scenario / Description of use case in detail	Send a GET-request to the endpoint of a cipher with following parameters: - Ciphertext - Key Receive the plaintext.
Post-Condition	User has a plaintext.
Exceptions	1: Loss of connection during the encryption process. 2: Incompatible Key size. Result: HTTP error code with message. 3: Wrong Key → Plaintext will be nonsense.

Table 4 Use Case 3

Hashing

Use Case #4	Hashing
Pre-Condition	None, User has a plaintext
Main Scenario / Description of use case in detail	Send a GET-request to the endpoint of a hashing algorithm with following parameter: - Plaintext Receive the hash.
Post-Condition	User has a hash.
Exceptions	Loss of connection during the encryption process.

Table 5 Use Case 4

Hashing with hash to compare

Use Case #5	Hashing
Pre-Condition	None, User has a plaintext and hash
Main Scenario / Description of use case in detail	Send a GET-request to the endpoint of a hashing algorithm with following parameter: - Plaintext - Hash Receive the hashed message, predefined hash, and if these are the same (true or false).
Post-Condition	User knows if the plaintext is the same as the hash.
Exceptions	Loss of connection during the encryption process.

Table 6 Use Case 5

4 UML Use Case Diagram

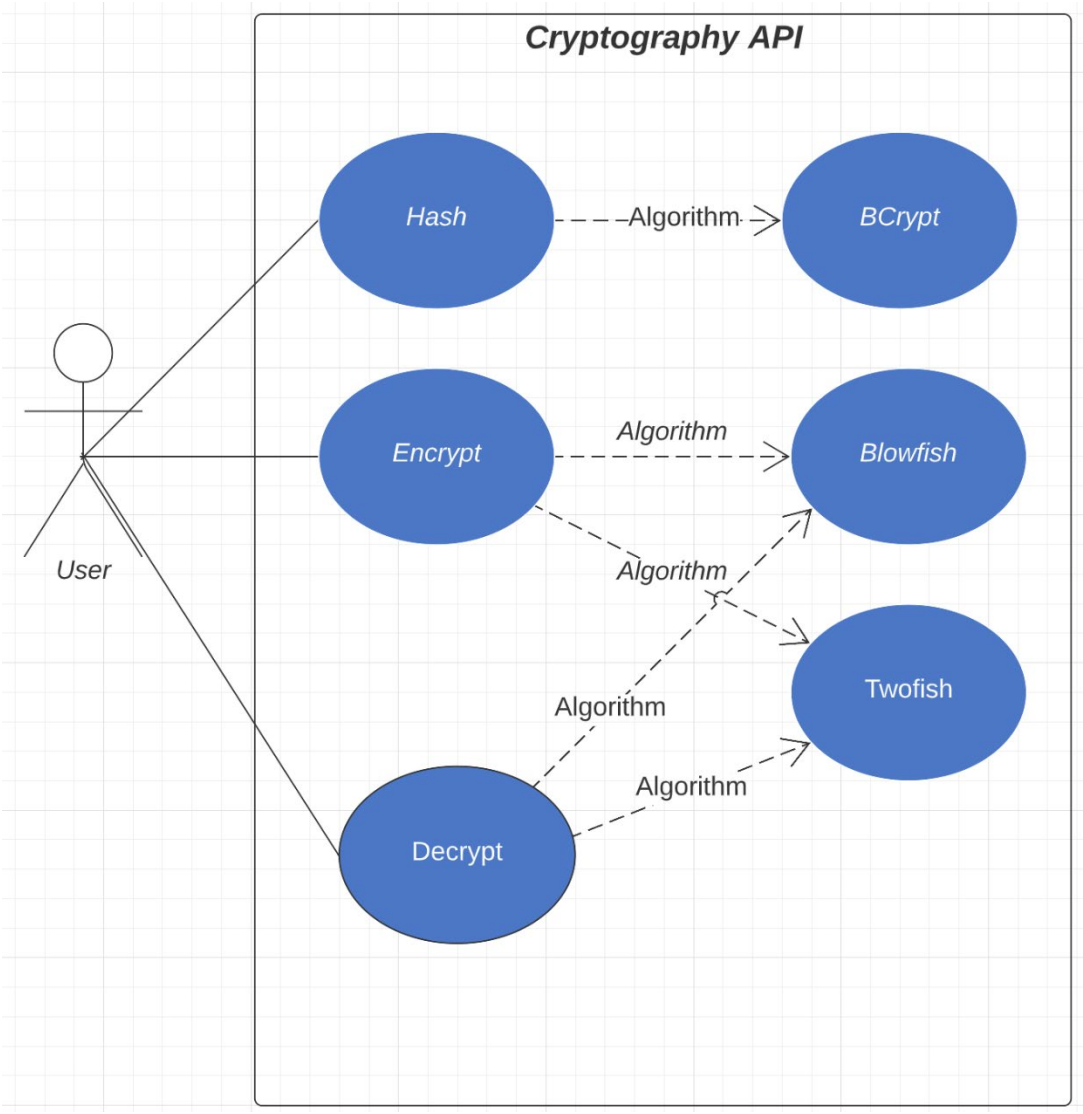

Figure 2 Use Case Diagram

5 UML Class Diagram

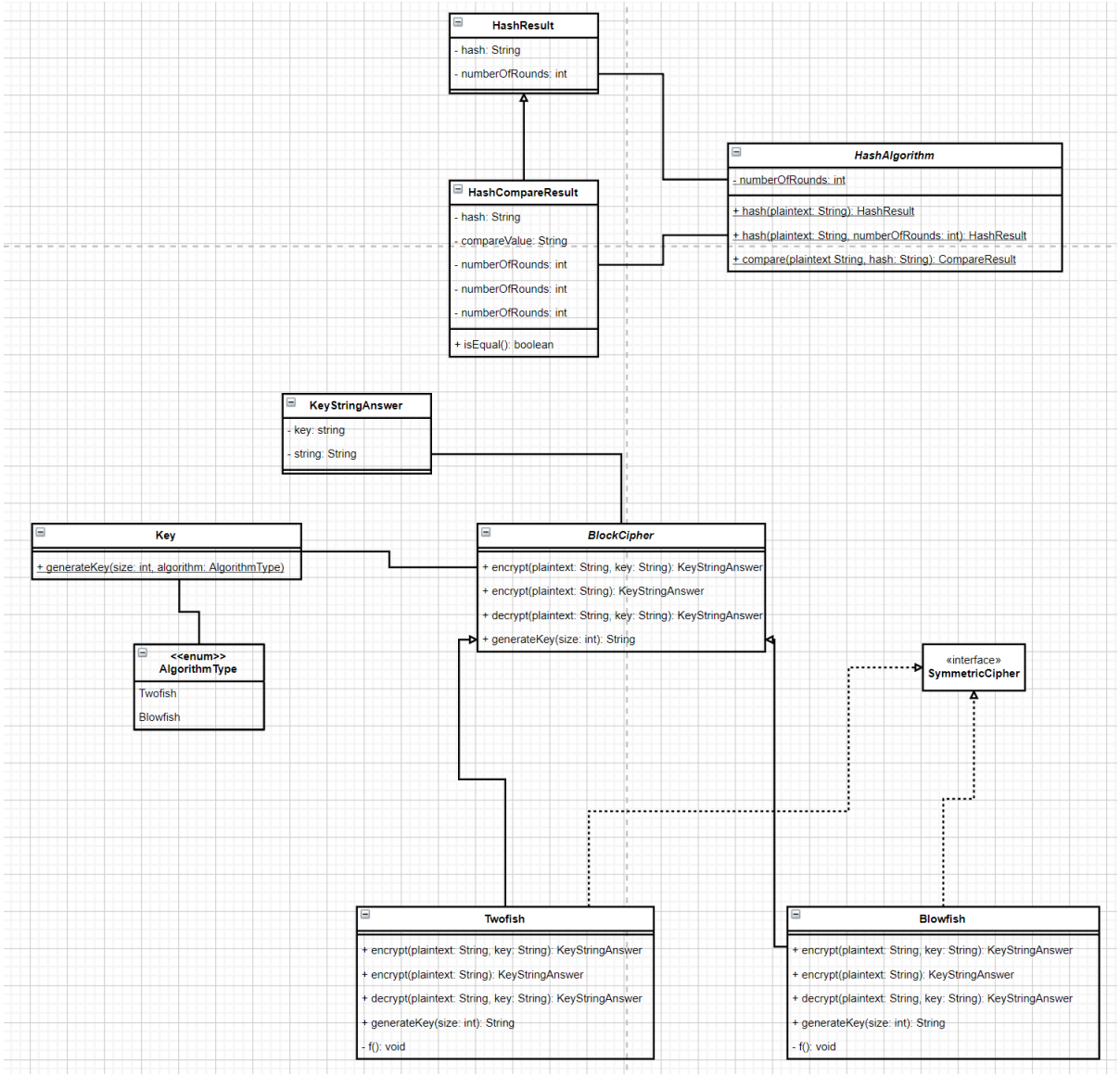

Figure 3 Class Diagram