|  |  |
| --- | --- |
| etf_logo_2010_cir | **UNIVERZITET U BEOGRADU**  **ELEKTROTEHNIČKI FAKULTET**  Zaštita podataka – 2020/2021  Bulevar kralja Aleksandara 73, PF 35-54, 11120 Beograd, Srbija |

OpenPGP Crypto GUI

Autori:

Vasilije Stambolić 0061/2018

Nemanja Maksimović 0355/2016

Sadržaj

[1. Uvod 3](#_Toc74254207)

[2. UI 4](#_Toc74254208)

[2.1. Glavni meni 4](#_Toc74254209)

[2.2. Enkripcija/potpisivanje 4](#_Toc74254210)

[2.3. Manipulacija ključevima 5](#_Toc74254211)

[2.4. Dekripcija 6](#_Toc74254212)

[3. Opis korišćenih algoritma 7](#_Toc74254213)

[3.1. Asimetrični algoritmi 7](#_Toc74254214)

[3.2. Simetrični algoritmi 7](#_Toc74254215)

[4. Opis klasa 8](#_Toc74254216)

[4.1. Key managment 8](#_Toc74254217)

[4.2. Transfer 10](#_Toc74254218)

[4.3. Exceptions 11](#_Toc74254219)

# Uvod

PGP (*Pretty Good Privacy*) je kriptografski protokol koji pruža privatnost (*enkripciju*) i autentikaciju (*potpisivanje*) u komunikaciji. PGP se koristi za potpisivanje, enkripciju i dekripciju mejlova, fajlova, tekstova da bi se povećala sigurnost u e-mail komunikaciji.

Cilj projekta je prikazati jednu implementaciju PGP protokola (OpenPGP) koristeći algoritme koji će biti opisani u nastavku dokumenta.

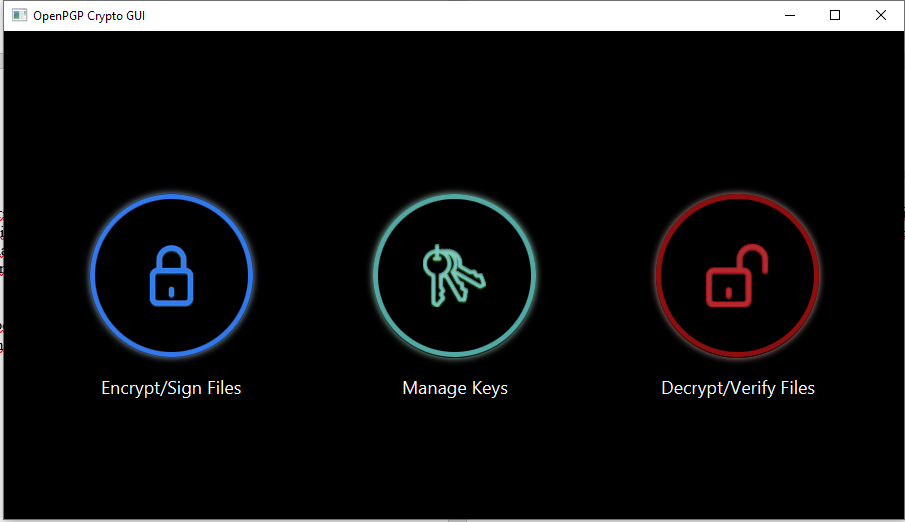
Glavne funkcionalnosti sistema su:

* Generisanje novog i brisanje postojećeg para ključeva
* Uvoz i izvoz javnih i privatnih ključeva
* Slanje poruke
* Primanje poruke

# UI

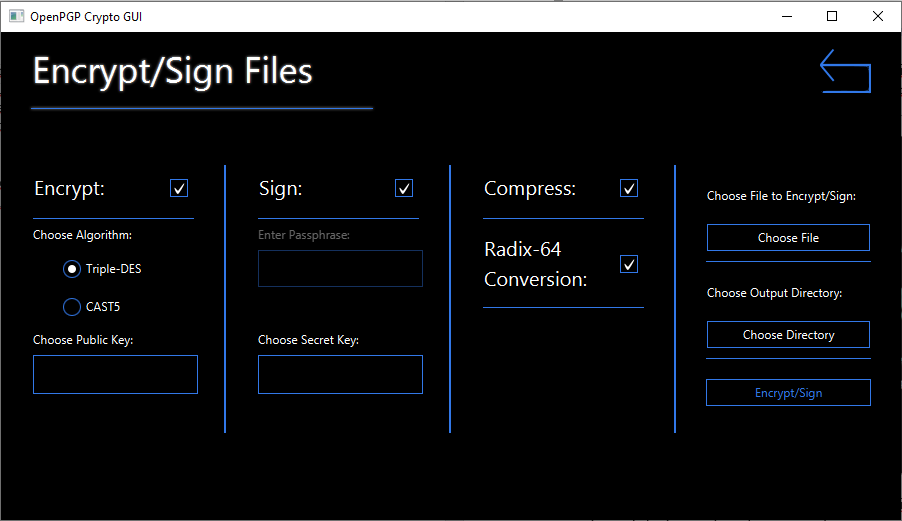
## Glavni meni

Korisnički interfejs se sastoji iz glavnog menija koji dalje vodi u tri posebne stranice: enkripcija, kreiranje i manipulisanje ključevima i dekripcija.



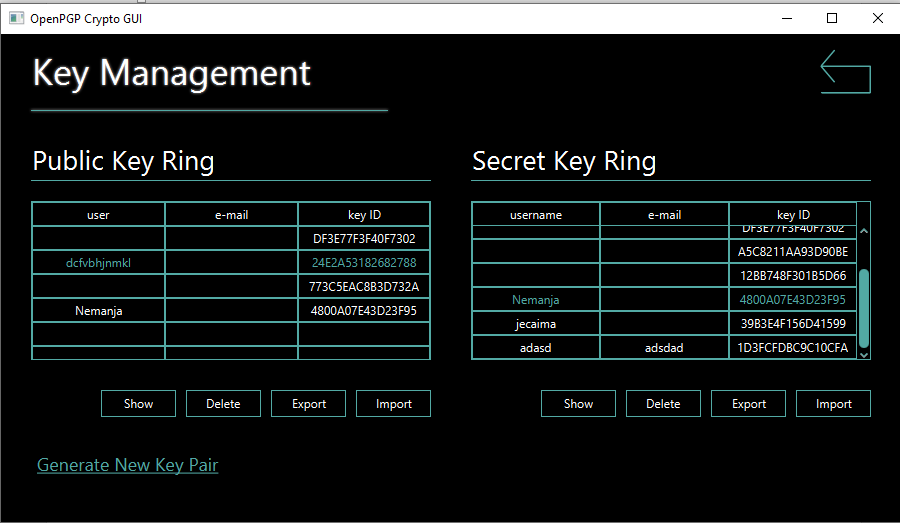
## Enkripcija/potpisivanje

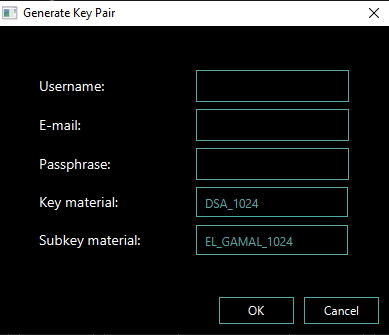
Stranica za enkripciju služi da se izabare fajl koji će da se enkriptuje i lokacija gde će on biti snimljen. Takođe na toj stranici se biraju svi potrebni parametric koji su potrebni za enkripciu/potpisivanje kao što su: Algoritam koji se koristi za enkripciju, javni i tajni ključ kao i opcije da li korisnik želi da odradi enkripciju, potpisivanje, kompresiju i konvertovanje u Radix64 format.



## Manipulacija ključevima

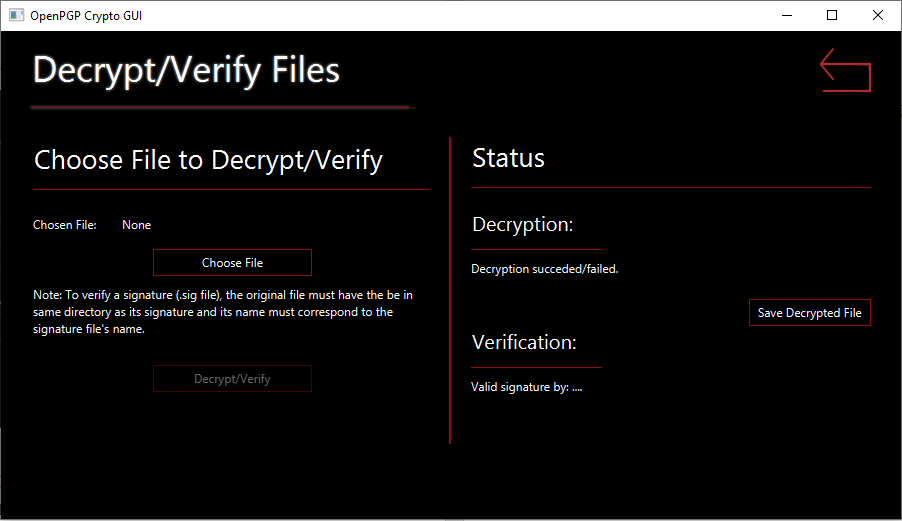
Na stranici za manipulaciju ključeva (Key Management) može da se doda novi par ključreva kao I da se obriše već postojeći. Pored ovih funkcionalnosti ključevi mogu da se eksportuju i importuju.





## Dekripcija

Na stranici za dekripciju i verifikaciju bira se fajl koji želimo da dekriptujemo ili verifikujemo kao i lokaciju gde fajl treba da se snimi.



# Opis korišćenih algoritma

Od algoritma za asimetrične ključeve aplikacija koristi DSA za potpisivanje i ElGamal za enkripciju.

Korišćeni algoritmi za simetrične ključeve su 3DES sa EDE konfiguracijom i CAST5.

## Asimetrični algoritmi

DSA (*Digital Signature Algorithm*) je public-key enkripcioni algoritam koji se koristi za generisanje elektronskih potpisa. Kod DSA kreira se public-private par ključeva koji se koriste za potpisivanje i autentikaciju. Pošiljaoc potpisuje podatke koje šalje svojim privatnim ključem što obezbeđuje jedinstveni potpis za korisnika dok primaoci javnim ključem pošiljaoca verifikuju od koga je poruka došla.

ElGamal algoritam je algoritam za enkripciju sa asimetričnim ključevima. Poruka se šifruje javnim ključem primaoca dok primaoc dekriptuje poruku svojim privatnim ključem. Time se obezbeđuje da samo primalac može da dekriptuje poruku. U PGP algoritmu ElGamal se koristi za enkripciju simetričnog ključa koji se koristi za enkripciju same poruke.

## Simetrični algoritmi

3DES algoritam je algoritam sa simetričnim ključem koji se koristi za enkripciju blokovskih podataka. 3DES je varijacija DES algoritma gde se sam algoritam ponovi tri puta. U ovoj konkretnoj implementaciji koristi se EDE konfiguracija (Encrypt-Decrypt-Encrypt) sa tri kluča.

CAST5 algoritam je algoritam sa simetričnim ključem koji se koristi kao podrazumevani algoritam u nekim verzijama PGPa. CAST5 koristi Feistel strukturu sa 12-16 rundi i ključevima od 40-128 bita (u konkretnoj implementaciji korišćeni su ključevi od 128 bita).

# Opis klasa

## Key managment

Klasa KeyInfo: abstraktna klasa koja čuva podatke potrebne za generisanje ključeva kao što su keyID username i email korisnika.

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| KeyInfo(long keyId, String userId) |  | Konstruktor koji na osnovu keyId i userId kreira nov objekat |
| getUsername() | String | Geter za atribut username |
| getEmail() | String | Geter za atribut email |
| getKeyIdLong() | long | Geter za atribut keyId |
| setKeyId(long keyId) | void | Seter za za atribut keyId |
| setUsername(String username) | void | Seter za atribut username |
| setEmail(String email) | void | Seter za atribut email |
| setUserInfo(String userId) | void | Na osnovu id korisnika postavlja podatke o korisniku (username, email) |
| formatKeyId(long keyId) | String | Konvertuje keyId u string |

Klasa PublicKeyInfo: nasleđuje KeyInfo. Koristi se za čuvanje podataka o javnom ključu

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| PublicKeyInfo(PGPPublicKey pgpPublicKey) |  | Konstruktor koji na osnovu PGPPublicKey objekta kreira novi objekat PublicKeyInfo |

Klasa SecretKeyInfo: nasleđuje KeyInfo. Koristi se za čuvanje podataka o tajnom ključu

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| SecretKeyInfo(PGPSecretKey pgpPublicKey) |  | Konstruktor koji na osnovu PGPSecretKey objekta kreira novi objekat PublicKeyInfo |

Klasa User: čuva podatke o korisniku

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| User(String username, String email, String passphrase) |  | Konstruktor koji inicijalizuje atribute novog objekta |
| getUsername() | String | Geter za atribut username |
| getEmail() | String | Geter za atribut email |
| getPassphrase() | String | Geter za atribut passphrase |
| getId() | String | Kreira id korisnika na osnovu usernam-a i emaila |

Klasa KeyManager: sadrži logiku za kreiranje, menjanje, brisanje, importovanje, eksportovanje ključeva unutar prstenova ključeva.

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| loadKeyRings() | void | Loads key rings from file |
| generateKeySubpacketVector() | PGPSignatureSubpacketVector | Generiše PGPSignatureSubpacketVector koji sadrži podatke o sertifikatu koji se koristi za potpisivanje |
| generateSubkeySubpacketVector() | PGPSignatureSubpacketVector | Generiše PGPSignatureSubpacketVector koji sadrži podatke o enkripciji asimetričnim ključem |
| generateKeys(User user, KeyMaterial keyMaterial, SubkeyMaterial subkeyMaterial) | Pair<PublicKeyInfo, SecretKeyInfo> | Generiše par ključeva koristeći prosleđene podatke |
| generateSubkeyPair(SubkeyMaterial subkeyMaterial) | KeyPair | Generiše par podključeva koristeći prosleđene parametre (za ElGamal) |
| generateKeyPair(KeyMaterial keyMaterial) | KeyPair | Generiše par ključeva koristeći prosleđene parametre (za DSA) |
| createKeyRingGenerator(User user, KeyPair keyPair, KeyPair subkeyPair) | PGPKeyRingGenerator | Kreira PGPKezRingGenerator  Za korisnika koristeći glavni i sporedni par ključeva |
| getPublicKeyInfoCollection() | Collection<PublicKeyInfo> | Vraća kolekciju javnih ključeva |
| getSecretKeyInfoCollection() | Collection<SecretKeyInfo> | Vraća kolekciju tajnih ključeva |
| exportSecretKeyRings() | void | Eksportuje kolekciju tajnih ključeva u fajl |
| exportPublicKeyRings() | void | Eksportuje kolekciju javnih ključeva u fajl |
| exportKey(KeyInfo keyInfo, File file) | void | Eksportuje konkretan ključ u fajl |
| importKeyRings(File file) | List<KeyInfo> | Importuje prsten ključeva iz prosleđenog fajla |
| getPublicKeyRing(KeyInfo keyInfo) | PGPPublicKeyRing | Dohvata prsten javnih ključeva kome pripada prosleđeni ključ |
| getSecretKeyRing(KeyInfo keyInfo) | PGPSecretKeyRing | Dohvata prsten tajnih ključeva kome pripada prosleđeni ključ |
| deletePublicKey(KeyInfo keyInfo) | void | Uklanja javni ključ |
| deleteSecretKey(KeyInfo keyInfo) | void | Uklanja tajni ključ |
| isEncrypted(KeyInfo keyInfo) | boolean | Proverava da li je prosleđeni tajni ključ zaštićen šifrom (passphrase) |

## Transfer

Klasa Sender: sadrži logiku za enkripciju i potpisivanje fajlova

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| Sender(File file, File outputDirectory, boolean compressionEnabled, boolean radix64Enabled, boolean encryptEnabled, int symmetricAlgorithmId, PublicKeyInfo publicKeyInfo, boolean signEnabled, String passphrase, SecretKeyInfo secretKeyInfo) |  | Konstruktor |
| send() | void | Sadrži svu potrebnu logiku za enkripciju, potpisivanje, kompresiju u zavisnosti od atributa klase |

Klasa Reciever: Sadrži logiku za dekriptovanje i verifikaciju fajlova

|  |  |  |
| --- | --- | --- |
| Potpis metode | Povratna vrednost | Opis |
| Receiver(File file) |  | Konstruktor |
| receive() | void | Sadrži logiku za dekripciju i verifikaciju u ondosu na sadržaj fajla (da li je fajl enkriptovan i/ili potpisan) |
| decrypt() | void | Dekriptuje fajl koristeći odgovarajući algoritam |
| decompress() | void | Vrši dekopresiju fajla ukoliko je bio kompresovan |
| read() | void | Čita sadržaj fajla |
| verify() | void | Verifikuje potpis fajla |

Klasa RecieverStatus: Klasa u koju se upisuje rezultat dekripcije/verifikacije. Sadrži dekriptovanu poruku kao i podatke o potpisivaču. Klasa ne sadrži metode sa posebnom logikom već samo getere i setere za atribute koji čuvaju potrebne podatke.

## Exceptions

Postoje nekoliko klasa koje nasleđuju Exception i služe za hendlovanje posebnih grešaka.

* InvalidFileFormatException : greška koja se prijavljuje ukoliko fajl koji je prosleđen na dekripciju nije u odgovarajucem formatu
* RecieverException : tip greške koji se javlja tokom dekripcije/verifikacije.

Klase koje nasleđuju RecieverException:

* + InvalidPassphraseException : javlaj se ukoliko passphrase koji je unet ne odgovara odabranom ključu
  + KeyNotFoundException : greška kada ključ korišćen za enkrpiciju ne postoji u odgovarajućem prstenu
  + PassphraseRequiredException : greška kada nije unet passphrase sa određeni ključ
  + SignerKeyNotFoundException : greška kada ključ korišćen za potpisivanje ne postoji u prstenu ključeva