Универзитет у Београду
Електротехнички факултет

Пројекат из Заштите података 2020/2021. година

Студенти:

Душан Стијовић 0145/2017

Филип Царевић 0065/2017

Садржај

О пројекту 3
Имплементирани алгоритами- DSA.....4
Имплементирани алгоритами- ElGamal 6
Референце.....8

О пројекту

Идеја пројекта јесте имплементација PGP сервиса:

- Увоз и извоз јавних/приватних кључева
  Кључеви се увозе из *.asc* формата. Аналогно, кључеви се извозе у исти формат.
- Креирање приватних кључева
  - За потписивање

    Подржано креирање *DSA* парова приватни/јавни кључ величине *1024* или *2048* бита.

    - За енкрипцију

    Подржано креирање *ElGamal* парова приватни/јавни кључ величине *1024*, *2048* или *4096* бита

- Дигитално потписивање
  Подржан Digital signature algorithm- *DSA* са кључевима величине *1024* и *4096* бита.
- Компресија
  Подржан *ZIP* алгоритам компресије.
- Шифровање
  Порука се шифрује једним од симетричних алогритама: *3DES* или *IDEA*.
  Сесијси кључ се генерише за сваку поруку. Приликом слања шифрује се *ElGamal* јавним кључем величине *1024*,*2048* или *4096* бита.
- Компатибилност са осталим мејл сервисима
  Подржана конверзија *radix64*.

Приликом имплементације коришћена је библиотека *Bouncy Castle.*

Имплементирани алгоритами- DSA

- Генерисање кључева:

Састоји се из 2 фазе. Прва фаза је генерисање параметара који су дељени између корисника, док се у другој фази генеришу сами кључеви.

Parameter generation [edit]

- Choose an approved cryptographic hash function H with output length $|H|$ bits. In the original DSS, H was always SHA-1, but the stronger SHA-2 hash functions are approved for use in the current DSS.[3][11] If $|H|$ is greater than the modulus length N, only the leftmost N bits of the hash output are used.
- Choose a key length L. The original DSS constrained L to be a multiple of 64 between 512 and 1024 inclusive. NIST 800-57 recommends lengths of 2048 (or 3072) for keys with security lifetimes extending beyond 2010 (or 2030).[12]
- Choose the modulus length N such that $N < L$ and $N \leq |H|$. FIPS 186-4 specifies L and N to have one of the values: (1024, 160), (2048, 224), (2048, 256), or (3072, 256).[3]
- Choose an N-bit prime q.
- Choose an L-bit prime p such that $p - 1$ is a multiple of q.
- Choose an integer h randomly from $\{2 \dots p-2\}$.
- Compute $g := h^{(p-1)/q} \mod p$. In the rare case that $g = 1$ try again with a different h. Commonly $h = 2$ is used. This modular exponentiation can be computed efficiently even if the values are large.

The algorithm parameters are (p, q, g). These may be shared between different users of the system.

Per-user keys [edit]

Given a set of parameters, the second phase computes the key pair for a single user:

- Choose an integer x randomly from $\{1 \dots q-1\}$.
- Compute $y := g^x \mod p$.

x is the private key and y is the public key.

- Генерисање потписа:

> A message m is signed as follows:
>
> - Choose an integer k randomly from $\{1 \dots q-1\}$
> - Compute $r := \left(g^k \bmod p\right) \bmod q$. In the unlikely case that $r = 0$, start again with a different random k.
> - Compute $s := \left(k^{-1}\left(H(m) + xr\right)\right) \bmod q$. In the unlikely case that $s = 0$, start again with a different random k.
>
> The signature is (r, s)

- Верификација потписа:

> One can verify that a signature (r, s) is a valid signature for a message m as follows:
>
> - Verify that $0 < r < q$ and $0 < s < q$.
> - Compute $w := s^{-1} \bmod q$.
> - Compute $u_1 := H(m) \cdot w \bmod q$.
> - Compute $u_2 := r \cdot w \bmod q$.
> - Compute $v := \left(g^{u_1} y^{u_2} \bmod p\right) \bmod q$.
> - The signature is valid if and only if $v = r$.

Имплементирани алгоритами- ElGamal

- Генерисање кључева:

> The first party, Alice, generates a key pair as follows:
>
> - Generate an efficient description of a cyclic group G of order q with generator g. Let e represent the unit element of G.
> - Choose an integer x randomly from $\{1, \ldots, q-1\}$.
> - Compute $h := g^x$.
> - The **public key** consists of the values (G, q, g, h). Alice publishes this public key and retains x as her **private key**, which must be kept secret.

- Енкриптовање података

> **Encryption** [edit]
>
> A second party, Bob, encrypts a message M to Alice under her public key (G, q, g, h) as follows:
>
> - Map the message M to an element m of G using a reversible mapping function.
> - Choose an integer y randomly from $\{1, \ldots, q-1\}$.
> - Compute $s := h^y$. This is called the *shared secret*.
> - Compute $c_1 := g^y$.
> - Compute $c_2 := m \cdot s$.
> - Bob sends the ciphertext (c_1, c_2) to Alice.
>
> Note that if one knows both the ciphertext (c_1, c_2) and the plaintext m one can easily find the shared secret s, since $c_2 \cdot m^{-1} = s$. Therefore, a new y and hence a new s is generated for every message to improve security. For this reason, y is also called an ephemeral key.

- Декритовање података:

Decryption [edit]

Alice decrypts a ciphertext (c_1, c_2) with her private key x as follows:

- Compute $s := c_1^x$. Since $c_1 = g^y$, $c_1^x = g^{xy} = h^y$ and thus it is the same shared secret that was used by Bob in encryption.
- Compute s^{-1}, the inverse of s in the group G. This can be computed in one of several ways. If G is a subgroup of a multiplicative group of integers modulo n, the modular multiplicative inverse can be computed using the Extended Euclidean Algorithm. An alternative is to compute s^{-1} as c_1^{q-x}. This is the inverse of s because of Lagrange's theorem, since $s \cdot c_1^{q-x} = g^{xy} \cdot g^{(q-x)y} = (g^q)^y = e^y = e$.
- Compute $m := c_2 \cdot s^{-1}$. This calculation produces the original message m, because $c_2 = m \cdot s$; hence $c_2 \cdot s^{-1} = (m \cdot s) \cdot s^{-1} = m \cdot e = m$.
- Map m back to the plaintext message M.

Референце

- *https://en.wikipedia.org/wiki/Digital_Signature_Algorithm*
- *https://en.wikipedia.org/wiki/ElGamal_encryption*