ALBANESE LAB

SOFTWARE ENGINEERING

GOST Toolkit

GOST Security Suite written in Go

Multi purpose cross-platform cryptography tool for symmetric encryption, cipher-based message authentication code (CMAC), recursive hash digest, hash-based message authentication code (HMAC), digital signature, shared key agreement (VKO) and PBKDF2 function for embedded systems.

GOST refers to a set of technical standards maintained by the Euro-Asian Council for Standardization, Metrology and Certification (EASC), a regional standards organization operating under the auspices of the Commonwealth of Independent States (CIS).

GOST is GOvernment STandard of Russian Federation (and Soviet Union):

- GOST 28147-89 64-bit block cipher (RFC 5830)
- GOST R 34.11-94 hash function 256-bit (RFC 5831)
- GOST R 50739-95 data sanitization method (non-cryptographic)
- GOST R 34.10-2001 public key signature function (RFC 5832)
- VKO GOST R 34.10-2001 key agreement function (RFC 4357)
- GOST R 34.10-2012 public key signature function (RFC 7091)
- VKO GOST R 34.10-2012 key agreement function (RFC 7836)
- GOST R 34.11-2012 Streebog hash function 256/512-bit (RFC 6986)
- GOST R 34.12-2015 128-bit block cipher Kuznechik (RFC 7801)
- GOST R 34.12-2015 64-bit block cipher Magma

Algorithms

Symmetric:

- Modes of Operation:
  - MGM: Multilinear Galois Mode (AEAD)
  - CTR: Counter Mode
  - OFB: Output Feedback Mode
- Block Ciphers:
  - GOST 28147-89 CryptoPro
  - GOST R 34.12-2015 Magma (default)

        - GOST R 34.12-2015 Kuznechik (Grasshopper)
    - Message Digest Algorithms:
        - GOST R 34.11-94 CryptoPro 256-bit
        - GOST R 34.11-2012 Streebog 256/512-bit (default)

Asymmetric:

- Public key Algorithms:
    - GOST R 34.10-2001 CryptoPro 256-bit
    - GOST R 34.10-2012 256/512-bit (default)
- Supported ParamSets:
    - GOST R 34.10-2001 256-bit: A, B, C, XA, XB
    - GOST R 34.10-2012 256-bit: A, B, C, D
    - GOST R 34.10-2012 512-bit: A, B, C

Features

- Cryptographic Functions:
    - Symmetric Encryption/Decryption
    - Digital Signature (ECDSA equivalent)
    - VKO shared key negociation (ECDH equivalent)
    - Recursive Hash Digest + Check
    - CMAC (Cipher-based message authentication code)
    - HMAC (Hash-based message authentication code)
    - PBKDF2 (Password-based key derivation function 2)
    - TLS (Transport Layer Security)
- Non-Cryptographic Functions:
    - GOST R 50739-95 data sanitization method
    - Bin to Hex/Hex to Bin string conversion
    - Random Art Public key Fingerprint (ssh-keygen equivalent)

Usage

```
usage of gosttk:
 -128
       Block size: 64 or 128. (for symmetric encryption only) (default 64)
 -512
       Bit length: 256 or 512. (default 256)
 -check string
       Check hashsum file. (- for STDIN)
 -cmac
       Compute cipher-based message authentication code.
 -crypt string
       Encrypt/Decrypt with symmetric ciphers.
 -derive
       Derive shared secret key (VKO).
 -digest string
       File/Wildcard to generate hashsum list. (- for STDIN)
```

```
-hex string
      Encode binary string to hex format and vice-versa.
-hmac
      Compute hash-based message authentication code.
-iter int
      Iterations. (for SHRED and PBKDF2 only) (default 1)
-key string
      Private/Public key, password or HMAC key, depending on operation.
-keygen
      Generate asymmetric keypair.
-mode string
      Mode of operation: MGM, CTR or OFB. (default "MGM")
-old
      Use old roll of algorithms.
-paramset string
      Elliptic curve ParamSet: A, B, C, D, XA, XB. (default "A")
-pbkdf2
      Password-based key derivation function 2.
-pub string
      Remote's side public key/remote's side public IP/PEM BLOCK.
-rand int
      Generate random cryptographic key: 128, 256 or 512 bit-length.
-recursive
      Process directories recursively. (for DIGEST command only)
-salt string
      Salt. (for PBKDF2 only)
-shred string
      Files/Path/Wildcard to apply data sanitization method.
-sign
      Sign with private key.
-signature string
      Input signature. (verification only)
-tcp string
      TCP/IP Transfer Protocol.
-verbose
      Verbose mode. (for CHECK command only)
-verify
      Verify with public key.
-version
      Print version information.
```

Examples

Asymmetric GOST R 34.10-2001 256-bit keypair generation (INI format):

```
./gosttk -keygen -old [-paramset A|B|C|XA|XB]
```

Asymmetric GOST R 34.10-2012 256/512-bit keypair generation (default):

```
./gosttk -keygen [-paramset A|B|C|D] [-512 -paramset A|B|C]
```

Signature (ECDSA equivalent):

```
./gosttk -sign [-512|-old] -key $prvkey < file.ext > sign.txt
sign=$(cat sign.txt)
./gosttk -verify [-512|-old] -key $pubkey -signature $sign < file.ext
```

VKO: Shared key negociation (ECDH equivalent):

```
/gosttk -derive [-512|-old] -key $prvkey -pub $pubkey
```

Encryption/decryption with Magma (GOST R 34.12-2015) symmetric cipher (default):

```
./gosttk -crypt enc -key $shared < plaintext.ext > ciphertext.ext
./gosttk -crypt dec -key $shared < ciphertext.ext > plaintext.ext
```

Encryption/decryption with Kuznyechik (GOST R 34.12-2015) symmetric cipher:

```
./gosttk -crypt enc -128 -key $shared < plaintext.ext > ciphertext.ext
./gosttk -crypt dec -128 -key $shared < ciphertext.ext > plaintext.ext
```

Encryption/decryption with GOST 28147-89 CryptoPro symmetric cipher:

```
./gosttk -crypt enc -old -key $shared < plaintext.ext > ciphertext.ext
./gosttk -crypt dec -old -key $shared < ciphertext.ext > plaintext.ext
```

CMAC-Kuznechik (cipher-based message authentication code):

```
./gosttk -cmac -128 -key $128bitkey < file.ext
```

CMAC-Magma (cipher-based message authentication code):

```
./gosttk -cmac [-old] -key $128bitkey < file.ext
```

GOST94-CryptoPro hashsum (list):

```

```

GOST94-CryptoPro hashsum (single):

```
./gosttk -digest - -old < file.ext
```

HMAC-GOST94-CryptoPro (hash-based message authentication code):

```
./gosttk -hmac -old -key $256bitkey < file.ext
```

Streebog256/512 hashsum:

```
./gosttk -digest - [-512] < file.ext
```

HMAC-Streebog256/512:

```
./gosttk -hmac [-512] -key $256bitkey < file.ext
```

PBKDF2 (password-based key derivation function 2):

```
./gosttk -pbkdf2 [-512|-old] -key "pass" -iter 10000 -salt "salt"
```

Note:

PBKDF2 function can be combined with the CRYPT, HMAC commands:

```
./gosttk -crypt enc -128 -pbkdf2 -512 -key "pass" < plaintext.ext > ciphertext.ext
./gosttk -hmac [-512] -pbkdf2 -key "pass" -salt "salt" -iter 10000 < file.ext
```

Shred (GOST R 50739-95 data sanitization method, 25 iterations):

```
./gosttk -shred keypair.ini -iter 25
```

Bin to Hex/Hex to Bin:

```
echo somestring|./gosttk -hex enc
echo hexstring|./gosttk -hex dec
```

TCP/IP Dump/Send:

```
./gosttk -tcp dump [-pub "8081"] > Pubkey.txt
./gosttk -tcp send [-pub "127.0.0.1:8081"] < Pubkey.txt
```

TLS Layer TCP/IP Dump/Send:

```
./gostls -tcp dump [-pub "8081"] > Pubkey.txt
./gostls -tcp send [-pub "127.0.0.1:8081"] < Pubkey.txt
```

Random Art (Public Key Fingerprint):

```
./gosttk -key $pubkey
```

License

Military Grade Reliability. Copyright (c) 2020-2021 ALBANESE Research Lab.

Source: *https://github.com/pedroalbanese/gosttk*
Download: *https://sourceforge.net/projects/gosttk*