

# Stream Cipher Algorithm: AES256GCM1K

AES256GCM1K is a chunked seekable cipher algorithm with integrity validation, based on AES-256-GCM. Due to additional integrity validation data, the size of the ciphertext exceeds the size of the plaintext by 32 bytes, plus 16 bytes for every full 1024-byte block of the plaintext. The asymptotic size overhead for a long stream is 16 / 1024 = 1.5625%.

## API access:

- Vault Item *meta.a*: string "AES256GCM1K"
- Vault Entry plaintext data, corresponding to the *encData* ciphertext:

 $\{32-byte AES key K\} \mid \mid \{12-byte seed S\}$ 

where "||" denotes concatenation.

#### Encrypting a Stream:

- Generate a random 16-byte value, assign it to G.
- Output *G* to the ciphertext stream.
- Calculate a hash

```
IVdata = sha256({12-byte S} || {16-byte G})
```

- Read a block of a maximum size of 1024 bytes from the plaintext stream, *L* is the size of the block. For the last block *L* is always less than the maximum, and might be 0.
- Encrypt the block using AES-256-GCM algorithm, with the key *K*, and the first 12 bytes of *IVdata* as the IV.
- Output the encrypted block to the ciphertext stream.
- Assign the 16-byte GMAC of the block to G.
- If *L* == 1024 bytes go to step 2.
- Otherwise output *G* to the ciphertext stream and finish.

## Decrypting a Stream:

- Read a 16-byte value from the ciphertext stream, assign it to *G*.
- Calculate a hash

IVdata = sha256({12-byte S} || {16-byte G})

- Read a block of a maximum size of 1040 bytes from the ciphertext stream, *L* is the size of the block. For the *L* is always less than the maximum.
- If L < 16 throw a ciphertext integrity error.
- Assign the last 16 bytes of the block to *G*.



- Decrypt the first (L 16) bytes of the block using AES-256-GCM algorithm, with the key *K*, and the first 12 bytes of *IVdata* as the IV.
- Validate GMAC against the value of *G*. If doesn't match throw a ciphertext integrity error.
- Output the decrypted block to the plaintext stream.
- If L == 1040 bytes go to step 2.
- Otherwise finish successfully.

# Seeking the stream to proceed with encryption or decryption from an arbitrary block boundary

- Seek operation allows to decrypt an arbitrary number of blocks from the middle of the ciphertext stream, or to overwrite the encrypted data starting from an arbitrary block boundary until the end of the stream.
- Based on the requested starting position, calculate the number of preceding full blocks *N*, 1024-byte blocks for the plaintext or 1040-byte blocks for the ciphertext.
- Seek the plaintext stream to  $N^*$  1024, seek the ciphertext stream to  $N^*$  1040.
- Read a 16-byte value of *G* from the ciphertext stream.
- Go to the step 2 of the encryption or decryption process respectively to proceed.
- Note: The ciphertext stream must be opened in the read-write mode to handle seekable encrypt operation.
- Note: If N == 0 for the encrypt operation G may be generated randomly and overwritten instead of reading it from the ciphertext stream.