# PSec

Sergey Strukov 2016, version 1.00

## 1. Introduction.

**PSec** is a data security protocol. It works on packet data channels. It provides encryption, authentication and anti-replay protection for the data packets. It also has the keep-alive capability and the close request function. The protocol uses the multi-key encryption, the key set is transparently refreshed during the packet communication. Each key has a limited life-time both by the work time and the number of encrypted octets. To start a communication a master key is required. The source of the master key is out of scope of the protocol. It may be some shared secret key, distributed by some means or it can be generated during some initial communication process (using some initial key exchange protocol).

## 2. Terms explanations.

*Byte aka octet* – an 8-bit quantity. Usually it is considered as an unsigned value with the value range {0, … , 255}.

*Packet* – a range of bytes. The main communication unit in packet data channels.

*Packet data channel* – a mean to exchange of packets between endpoints. The main example is the **UDP** protocol over an **IP** network.

*Packet endpoint* – a source or a destination of a packet transfer in a packet data network. Usually each endpoint has two associated packet length limits, one for inbound packets and the second for outbound packets. From the user perspective a packet endpoint has two attributes: max inbound packet length, max outbound packet length and two functions: send a packet and receive a packet.

***PSec** protocol entity* – is a protocol processing entity. It owns some packet endpoint and provides a packet endpoint for usage. The inner endpoint is insecure and the outer is secure. The outer endpoint has less packet length limits then the inner. A **PSec** protocol entity is initialized with a *master key*. Once two such entities are created on the two connected endpoints with the same master key, they provide a secure mean of communication, the new packet data channel. The packets withing this channel are protected from the ear-dropping, modification, fabrication and repeating, which may happens on the inner data channel.

*Block cipher* – is a function, this function processes a range of bytes into another range of bytes. Both ranges have the same length, this length is a cipher attribute and called the *cipher block length*. Another attribute is a *key length*. The cipher has a parameter. This parameter is called the *key*. This key must be set up before the cipher can work. The key itself is a range of bytes of the key length. The byte range transformation of the cipher is invertible. The inverse function is called the *inverse cipher*. It is also a block cipher with the same attributes. If you apply a cipher to a byte range and the inverse cipher to the result you will get the original byte range. Both direct and inverse ciphers must be set up with the same key. The process of the applying of a cipher is commonly called the encryption and the process of the applying of the inverse cipher is commonly called the decryption. The most popular ciphers are **AES** ciphers. There are three of them: **AES-128** with the key length 16, **AES-192** with the key length 24 and **AES-256** with the key length 32. The block length is 16 for all of them. **PSec** may use a cipher with the key length from 16 to 512 bytes and the block length from 8 to 64 bytes.

*Hash function* – is a function, this function take as input a range of bytes of any length and produces an output. The output is a range of bytes of the fixed length. This length is the hash function attribute and is called the hash (or digest) length. The commonly used in applications family of hash functions is **SHA** functions. There are following variants of such functions: **SHA-1**, **SHA-224**, **SHA-256**, **SHA-384**, **SHA-512**. They correspondent hash lengths are: 20, 28, 32, 48, 64.

**PSec** may use a hash function with the hash length from 16 to 64.

Diffie-Hellman group.

Key generator.

Random generator.

Life-time.

Key-set.

Convolution coder.

Keep-alive.