Secure Update Protocol

UNDER CONSTRUCTION

Overview

This secure protocol primary target is to program internal/external memories on final platform with SFL(or equivalent).

Secure Update Protocol

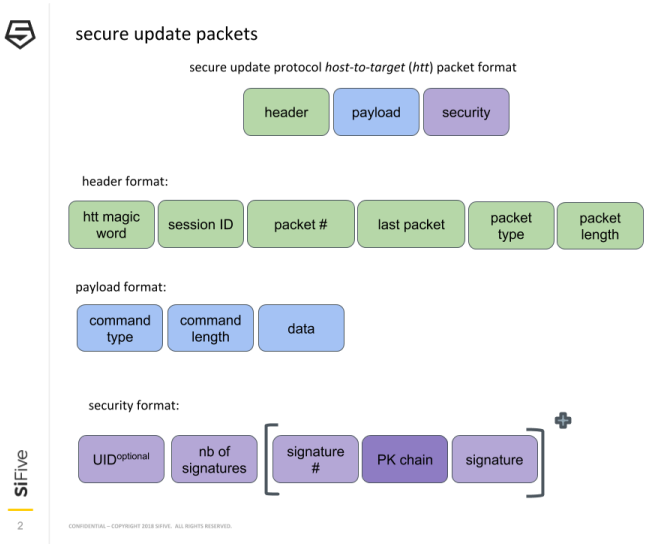
In the following document, Secure Update Protocol for RISC-V is denominated as SUP.

Specifications

Hereafter a list of features which are specification basis:

- This protocol is presented as "*secure*" because packets are covered with signature(s) and therefore cannot be accepted/processed if signature(s) is/are absent/corrupted/incoherent.
- This protocol is divided in 2 "layers":
  - "*Physical layer*" represented by packet abstraction.
  - "*Logical layer*" represented by command abstraction.
- This protocol enables implicitly the use of piece of code that is downloaded into internal RAM and executed from it. This feature is denominated "*applet*" in this document.
- This protocol cannot address directly external memories but using *applet*.
- Protocol allows to download multiple *applets* and/or execute it/them several times.
- A SUP packet is composed of one header, one payload, one security block and one ECC block.
- One communication session uses one session ID, using one/several packets, holding one command per packet.
- Session closure induces SBR to reset platform.
- Host sends next packet if, and only if it has previously received a communication acknowledgement from platform for previous packet treatment.
- SUP session always uses STK, never SSK/CSK when platform's life cycle is **Phase#0**.
- SUP session always uses SSK/CSK, never STK when platform's life cycle is **Phase#1**.
- SUP is inactive when platform's life cycle is **Phase#2**.
- SUP session always uses SSK/CSK and UID, never STK when platform's life cycle is **Phase #U**.
- When CUK is present in OTP, then SSK is no more used.
- Packet from platform to Host are only transmission acknowledgements with return error value. Except when platform performs *applet* execution; then it may answer back one/several response packets **after** last transmission acknowledgement.

Packet Format



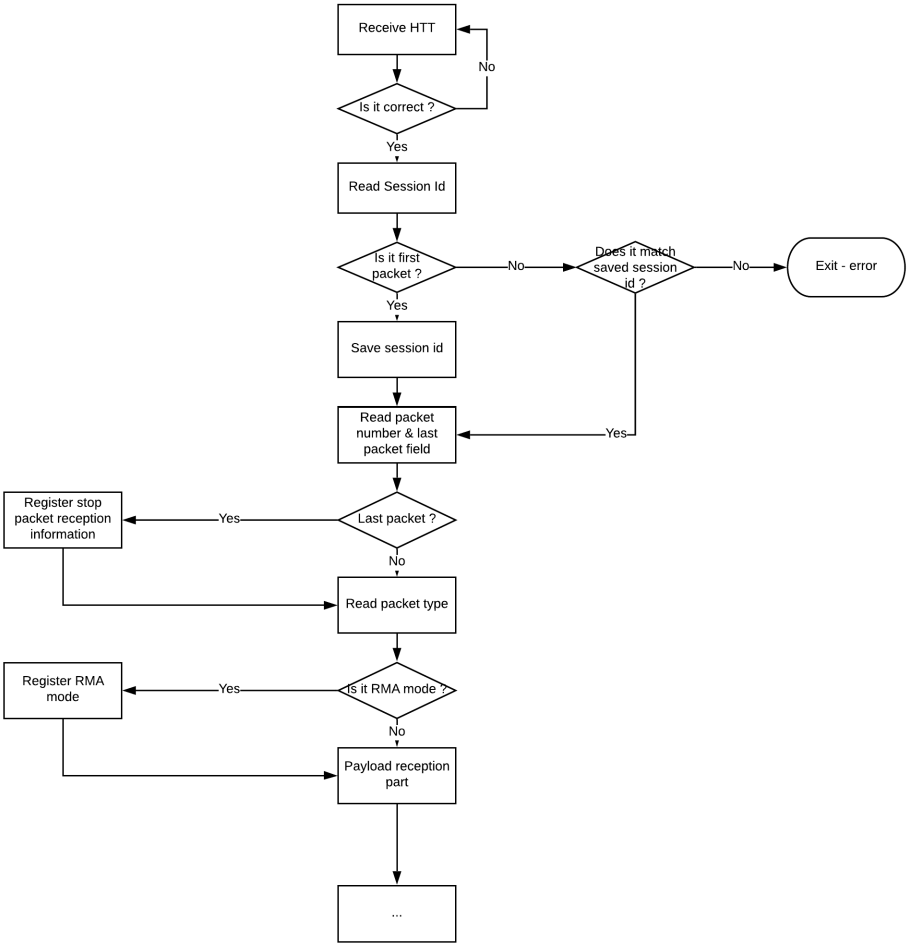
Header / Packet Layer

This "*layer*" can be considered as the "*transport layer*", meaning that it's where header and security check$_{(signature)}$ are processed.

- The '**HTT Magic Word**' - 4 Bytes - is a synchronization pattern - **0xAA51F17E**. It's used to trigger start of packet in incoming characters. SRC should loop$_{(with\ fixed\ characters\ depth)}$ until pattern is received. When characters depth is triggered, then process exits with error, leading to SLB step.
- The '**Session ID**' - 4 Bytes - given by Host must be constant during the whole communication session. This value range is '0x00000000' to '0xffffffff', with no specific value. If it varies, then process exits with error, leading to SLB step.
- The '**Packet Number**' - 4 Bytes - is a monotonic counter, incremented with each packet, form '0x00000000'$_{(start\ value)}$ to '0xffffffff'.
- The '**Last Packet**' - 4 Bytes - When this field is set to specific pattern - **0x1A574AC8** -, It indicates that current packet$_{(whatever\ its\ value\ is)}$ is the "*last to receive packet*". SBR does not expect any other packet after the current one; as a consequence, characters RX window is closed after receiving security part. If incoherence is pointed on this number, then process exits with error, leading to SLB step. For instance, if the communication has 4 packets, they are numbered 0, 1, 2 with '**Last Packet**' at any value$_{(but\ specific\ one)}$ ... and last 3 and *0x1A57 4AC8* as '**Last Packet**'.
- The '**Packet Type**' - 4 Bytes - field determines if current packet is "*Generic*" or "*RMA*". For this latter, the field UID is then relevant, meaning that value is checked with part's one. If it does not match, packet is rejected resulting in SUP end with error. When platform is in RMA mode, SBC only accepts RMA type packets.
- The '**Packet Length**' - 4 Bytes - field gives the number of Bytes coming after the header. It includes payload and security part.

There is no rebroadcast for SUP packet if there's an error.

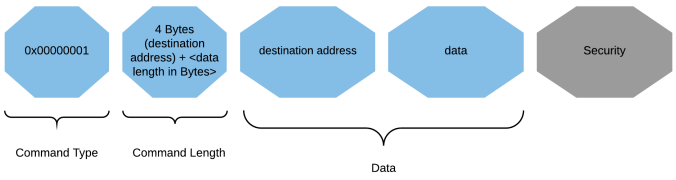
Packet Header Algorithm



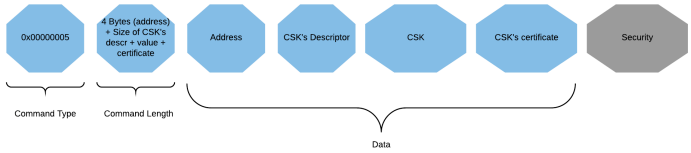
Payload / Command Layer

This "*layer*" can be considered as the "*logical layer*", meaning that it's where command is porcessed.

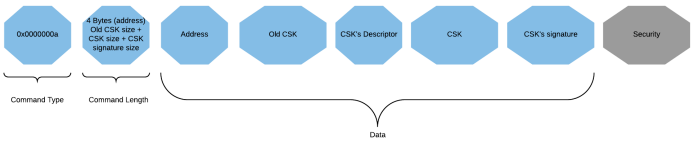
- The '**Command Type**' - 4 Bytes - field must be constant during the session. It gives the total number of segments expected for the whole communication session. It **does not** indicate the number of segments within the current packet. Value goes from '1' to '0xffffffff'. If it varies, then process exits with error, leading to SLB step.
- The '
- '**Command Length**' - 4 Bytes - field gives the size of "payload" in this packet. Therefore, it is specific to each packet. If this size is incoherent, the process exits with error, leading to SLB step.
- The fields hereafter represents **one command.**
    - Hereafter are shown each command with its format. It can be:
        - '**write-data**' - **0x00000001** - copies '**Data**' to '**Address**'.



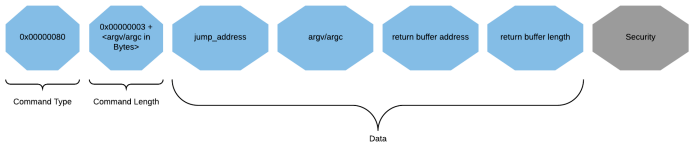
        - '**write-csk**' - **0xa94f2cb5** - this very particular command extract CSK and its signature form '**Data**' to program both into OTP at dedicated location only if all CSK's slots are free.



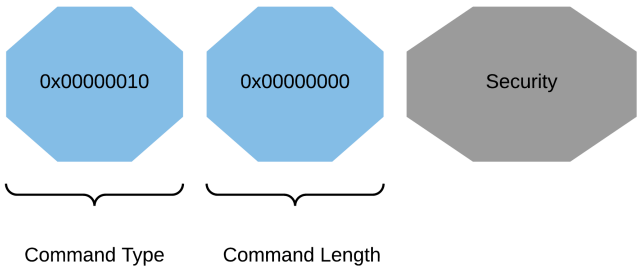
- '**update-csk**' - **0x68234fba** - performs same procedures as '**write-csk**' but when at least one slot is free$_{\text{(but not all)}}$.



- '**execute**' - **0x1e5dd280** - jumps into code located at '*jump_address*', gives '*argv/argc*' to code to be executed, indicates$_{\text{(if relevant)}}$ to SBR, address and length of data to be sent back as a result of applet processing. Note that a session can have several '**execute**' segment. They're executed sequentially,'; and an applet response is sent after each '**execute**' process.



- '**get-info**' - **0xc3788d10** - it answers the SBC version number, the chip UID, the RAM section available for the applets, the OTP CSK last slot index.

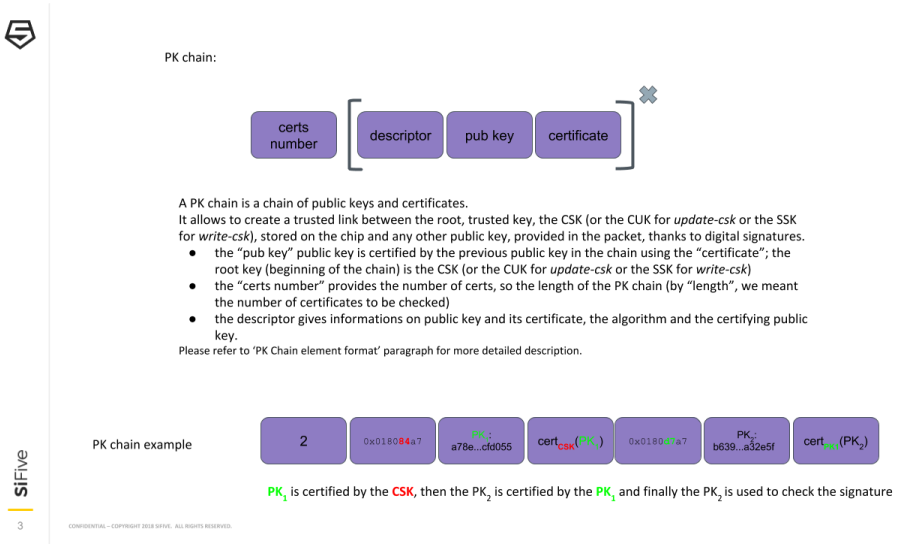


Security

The "security" is composed of:

- UID$_{\text{(optional)}}$: 16 Bytes - This is the chipset serial number. This field is only present in packet when session is 'RMA' not generic.
- **Number of signature(s)**: 4 Bytes - SBR supports multiple signatures, but all ultimately referring to CSK.
- **Signature number**: 4 Bytes - value starts from '0' to ('**Number of signature(s)**' - 1). Note that this signature identifier must be monotonic.
- **PK chain**$_{\text{(optional)}}$: m Bytes - It represents a certificates chain. Root is CSK.
- **Signature**: 96 Bytes - an ECDSA secp384r1 signature depending on previous PK Cain.

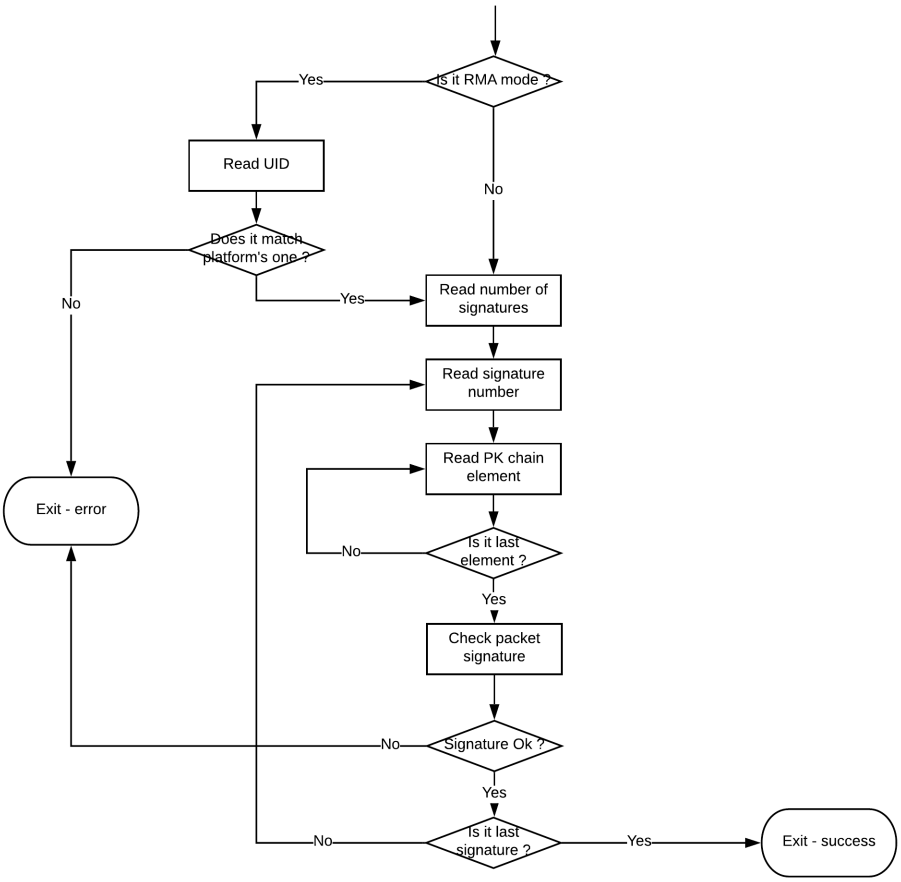
'**PK Chain**' structure is shown below:



Note that for future use:

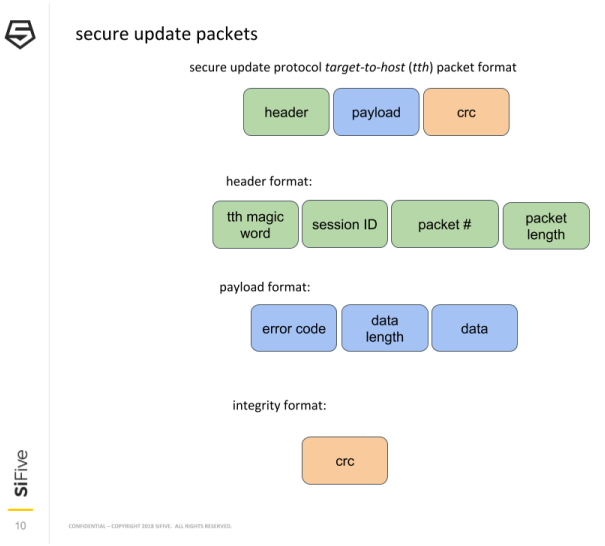
- Signature can use other curves(for ECDSA) or other algorithms(e.g. RSA).
- PK chain can use different keys lengths(different curves, greater or equal to secp384r1).

Security Algorithm



SUP Response Packet

This packet is sent by SBC *to Host* in response to packet sent *from Host*.



This packet is composed with:

- **TTH Magic Word**: 4 Bytes - This is a synchronization pattern. **0x5551F17E**
- **Session ID**: same during session communication.
- **Packet number**: same as previous incoming packet.
- **Packet Length**: covers '**Error Code**' + '**Data Length**' + '**Data**'*(if any)* and '**CRC**' fields. Minimum value is then 0x00000008.
- **Error Code**: 4 Bytes - Hereafter is shown "*No Error*" and "*Segment Error*" related values..
  - *0x00000000*: No Error..
  - 0x.....: Error code coming from SBC.
- **Data Length**: gives size of '**Data**' field in Bytes. This field is always present in response; if no data has to be transmitted, then its value is '0'.
- **Data**: field is present only if '**Data Length**' is not '0'. Relevant only for '*GET-INFO'* and *'EXEC UTE'* return data if any.

Note that a specific response packet is returned by SBC after applet execution if this latter has to send data back to Host.

Again, one *applet* response corresponds to one '**execute**' segment$_{\text{(sequentially)}}$.

Applet

An applet is a procedure executed in SBC's context. Meaning that, even if it has virtually access to the whole platform, it must not interfere with existing data in iRAM, used IPs$_{\text{(except if it's applet goal)}}$, ⋯

Format

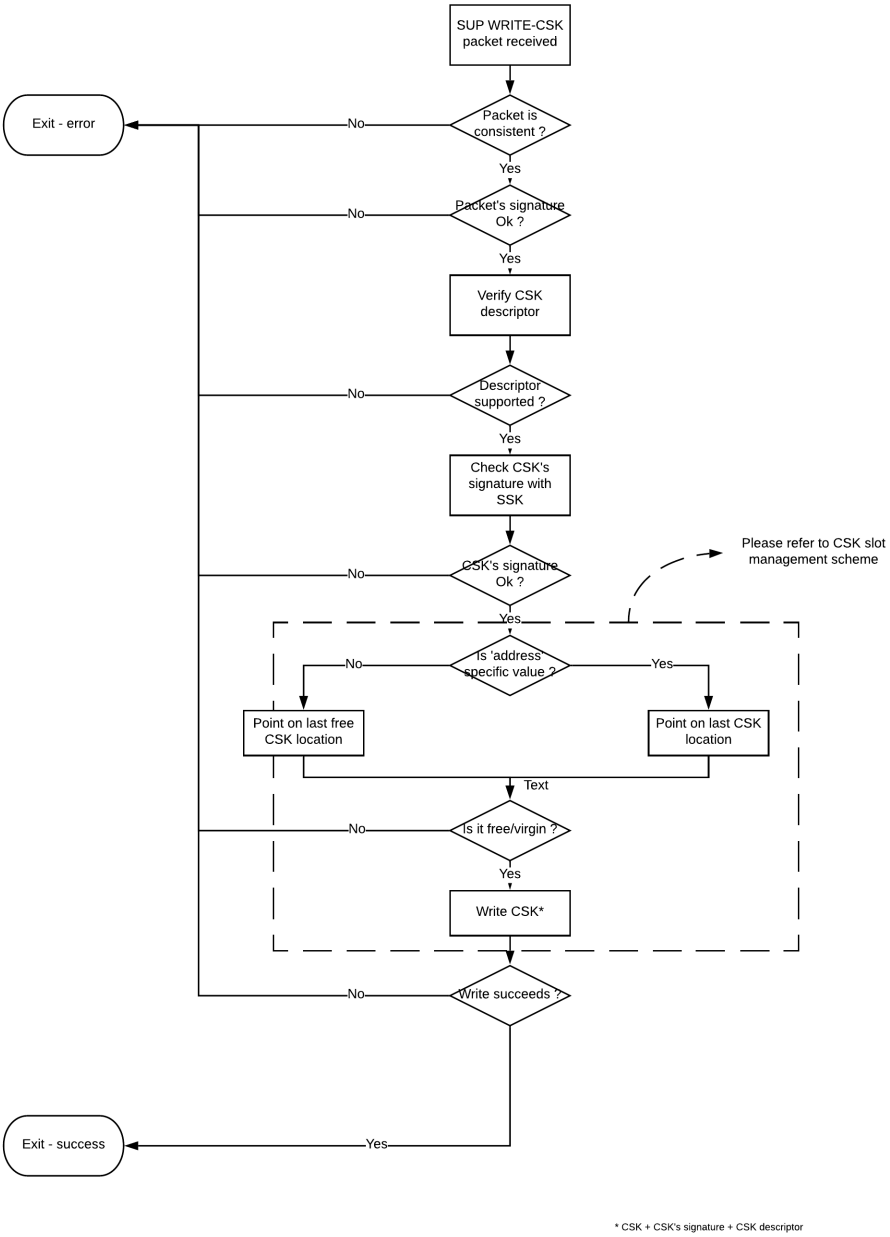
There is no specific format as for SLB or "*patch*". Its "**execute**" type segment that initiates applet execution with embedded parameters.

Applet always returns an "error" value to SBR but it can send back to Host a response buffer. This response buffer$_{\text{('Data' field)}}$ is managed by SUP as extra data in SUP response packet

Usecases

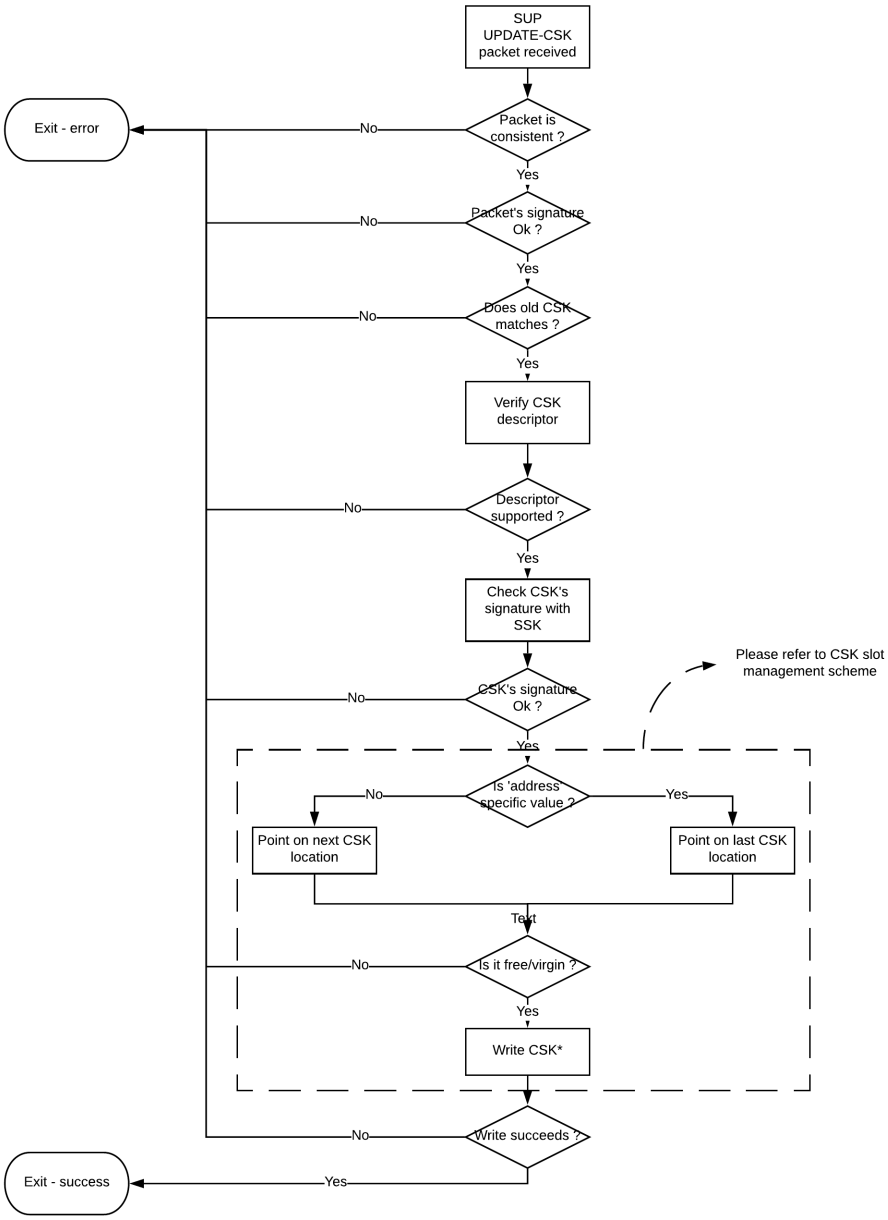
WRITE-CSK - nominal case

For this usecase, it starts after SUP packet reception and its check Ok.



UPDATE-CSK - nominal case

For this usecase, it starts after SUP packet reception and its check Ok.



* CSK + CSK's signature + CSK descriptor