

Integration of IPsec in Virtio-Net

Srujana Challa, Nithin Kumar Dabilpuram

Overview of IPsec(Internet Protocol Security)

Features:

- Data Confidentiality
- Data Integrity
- Authentication
- Replay Protection

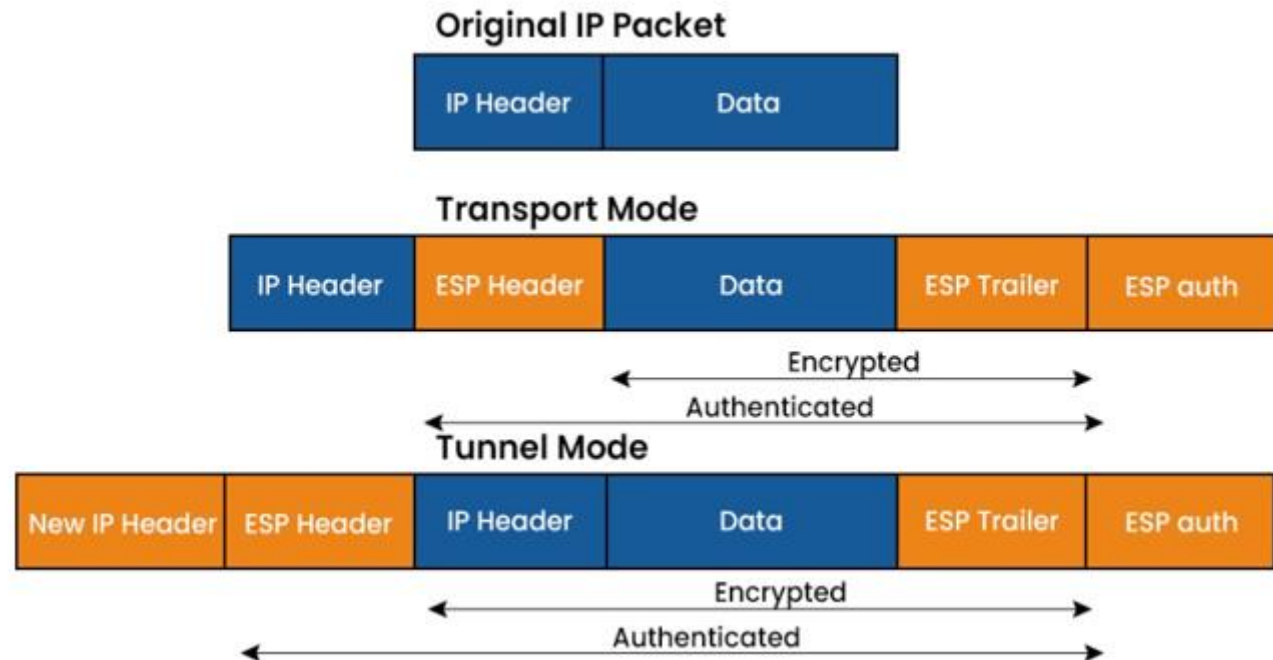
Protocols:

- ESP
- AH

Modes Of Operation:

- Tunnel
- Transport

IPsec ESP protocol Example:



Introduction

Config Support

- Capabilities
- Resource Objects

Flow Filter Actions

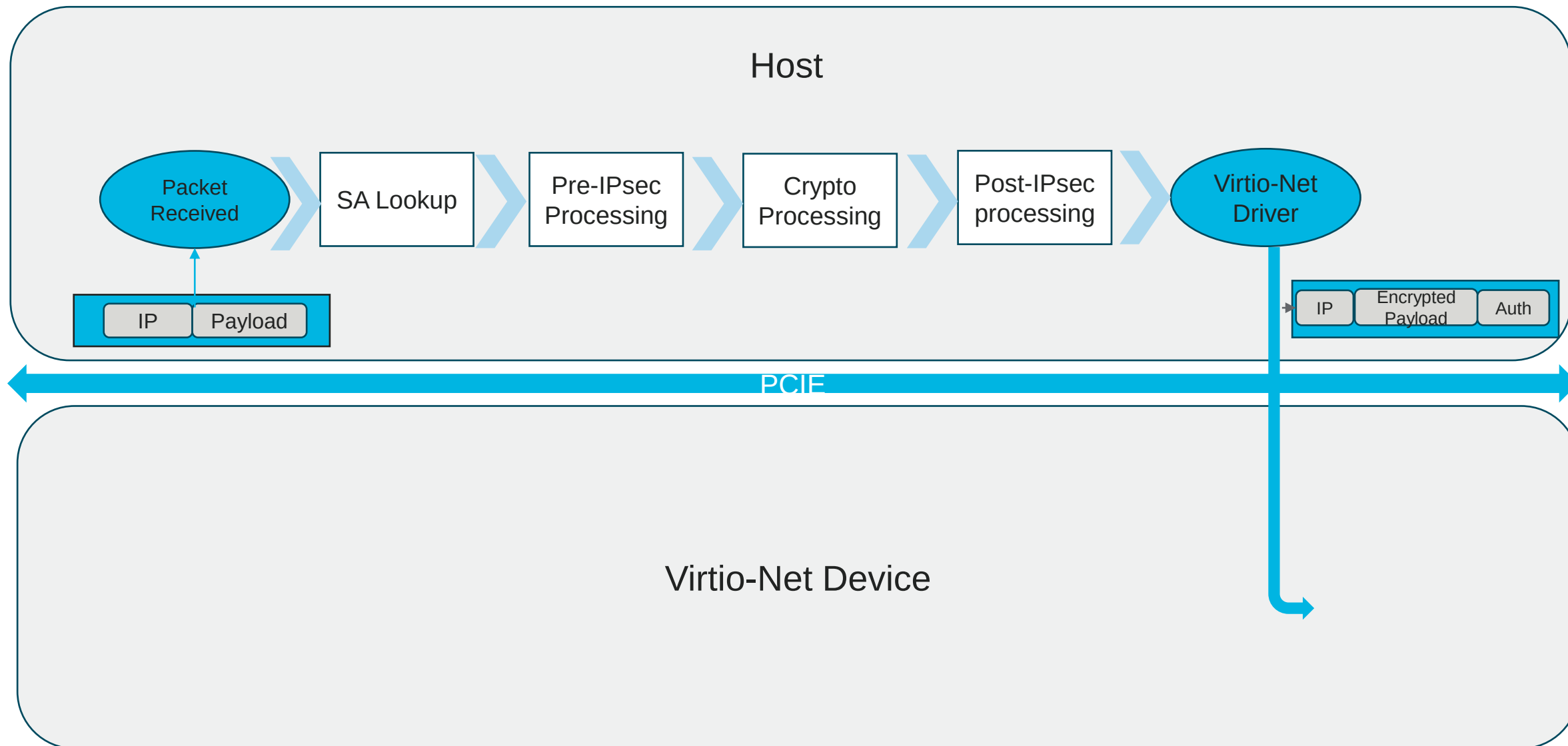
Introduce actions

- VIRTIO_NET_FF_ACTION_IPSEC
- VIRTIO_NET_FF_ACTION_IPSEC
_RECIRCULATE

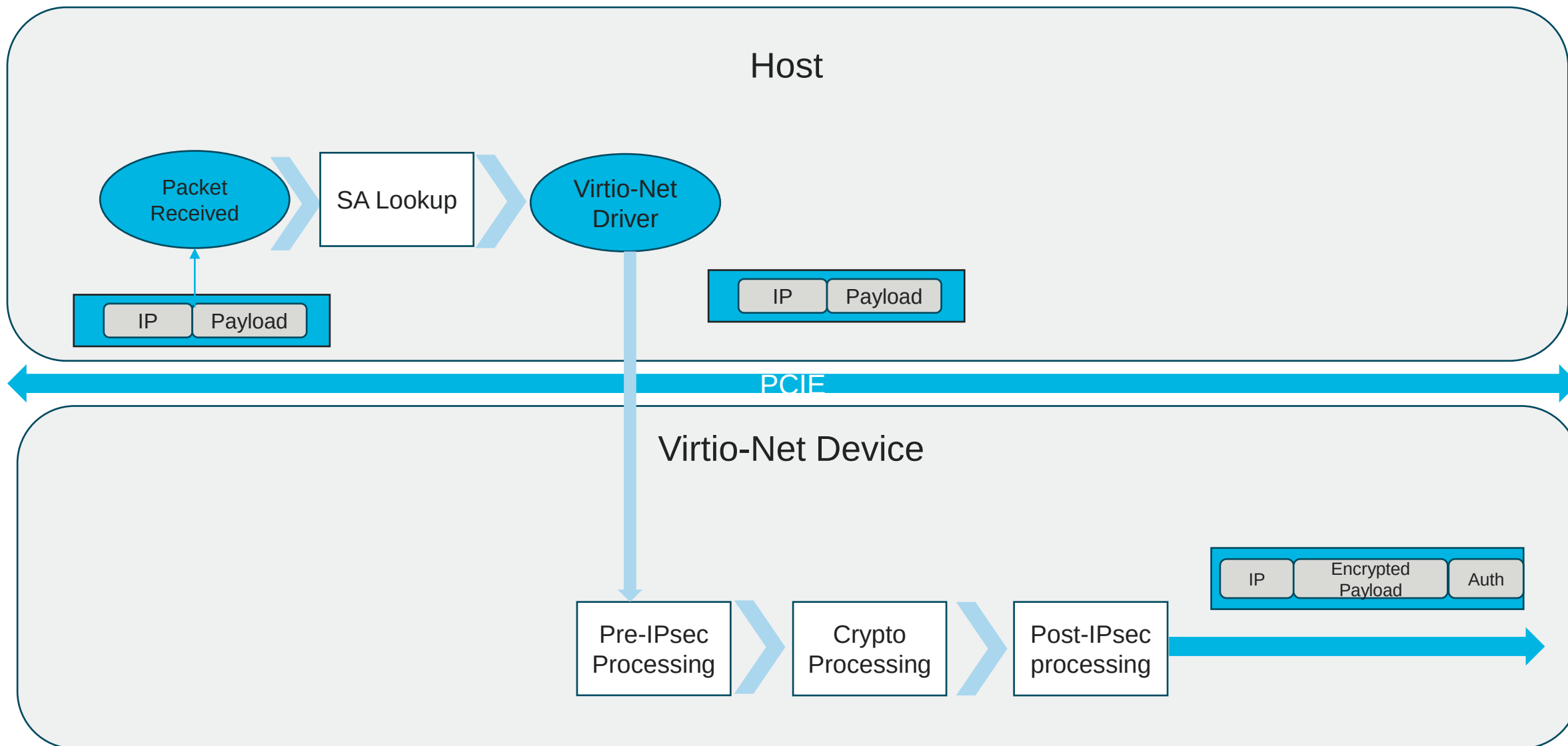
Struct Updates

- Updates to struct virtio_net_hdr

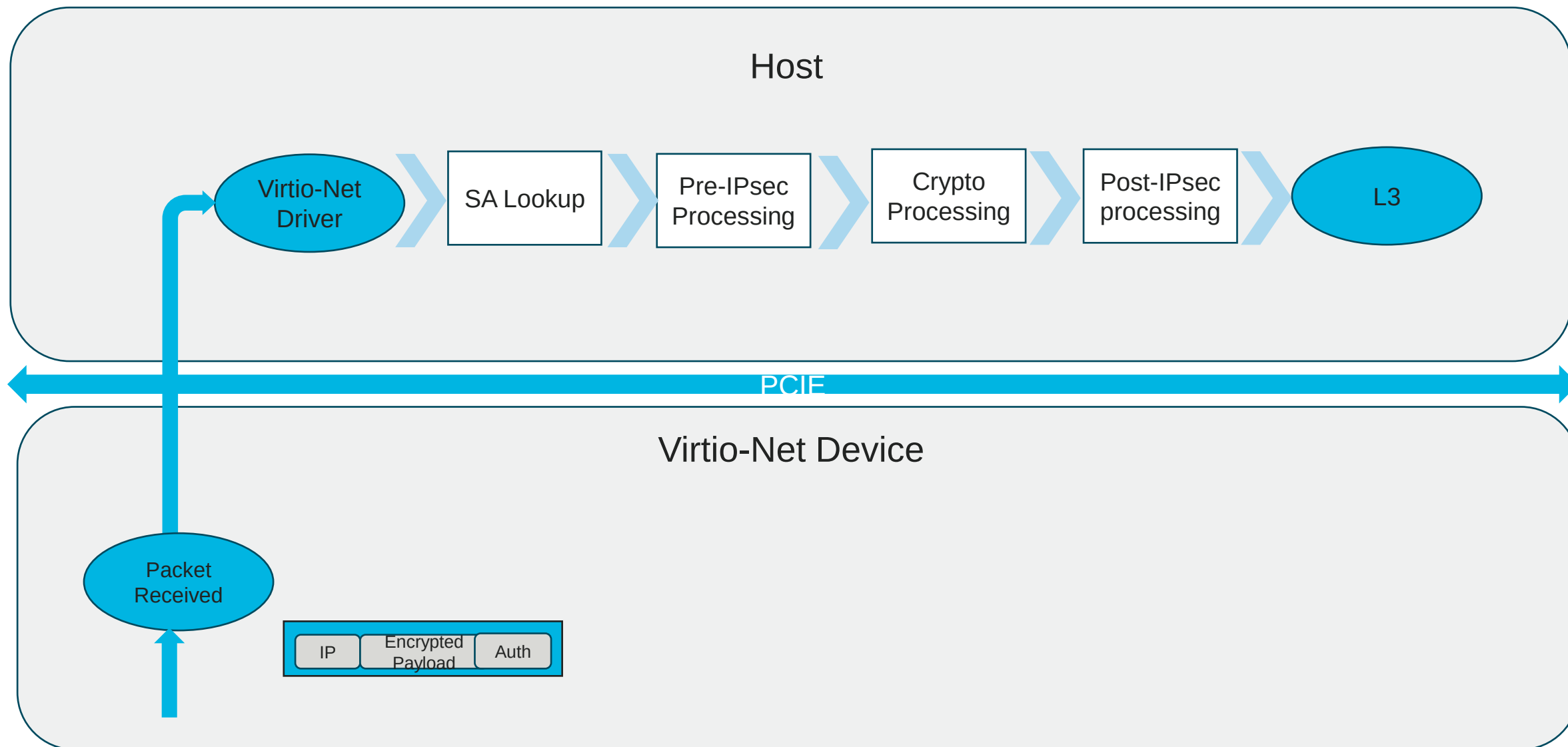
Outbound IPsec With Host Crypto



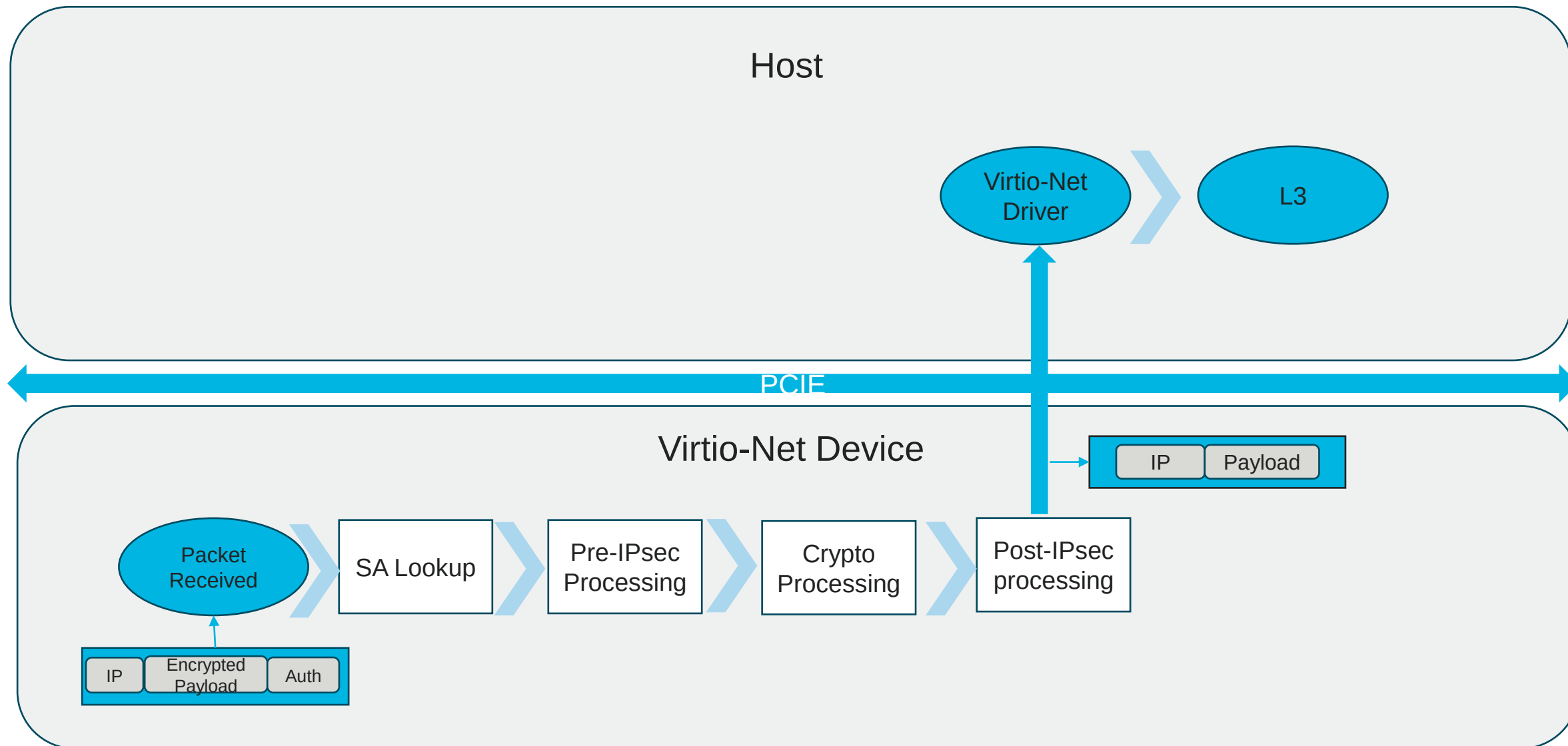
Outbound IPsec In Virtio-Net Device



Inbound IPsec With Host Crypto



Inbound IPsec In Virtio-Net Device



Advantages

Reduce CPU usage

Enhance Security

Device Compatibility

Future-Ready

Virtio-Net IPsec Design

VIRTIO_ADMIN_CMD_CAP_GET

```
struct virtio_admin_cmd_cap_get_data {  
    le16 id;  
    u8 reserved[6];  
};
```

VIRTIO_ADMIN_CMD_CAP_SET

```
struct virtio_admin_cmd_cap_set_data {  
    le16 id;  
    u8 reserved[6];  
    u8 cap_specific_data[];  
};
```

VIRTIO_ADMIN_CMD_RESOURCE_
OBJ_CREATE

```
struct virtio_admin_cmd_resource_obj_cmd_hdr {  
    le16 type;  
    u8 reserved[2];  
    le32 id; /* Indicates unique resource object id per resource object type */  
    le64 flags;  
    u8 resource_obj_specific_data[];  
};
```

VIRTIO_ADMIN_CMD_RESOURCE_
OBJ_DELETE

IPsec Capabilities

ID	Name	Description
0x803	VIRTIO_NET_IPSEC_RESOURCE_CAP	IPsec resource limits capability
0x804	VIRTIO_NET_IPSEC_SA_CAP	IPsec SA capability

VIRTIO_NET_IPSEC_RESOURCE_CAP

```
struct virtio_crypto_ipsec_resource_cap {  
    le32 inb_sa_limit;  
    le32 outb_sa_limit;  
};
```

VIRTIO_NET_IPSEC_SA_CAP

```
struct virtio_crypto_ipsec_mode_cap {  
    u8 mode;  
    u8 reserved[3];  
    le32 max_replay_win_sz;  
    le32 options;  
    le64 cipher_algo;  
    le64 hmac_algo;  
    le32 aead_algo;  
    le32 max_cipher_key_len;  
    le32 max_auth_key_len;  
};  
struct virtio_crypto_ipsec_sa_cap_data {  
    u8 count;  
    u8 reserved[7];  
    struct virtio_crypto_ipsec_mode_cap mode[];  
};
```

- ESP Transport
- ESP Tunnel
- AH Transport
- AH Tunnel

IPsec Resource Objects

Type	Name	Description
0x0203	VIRTIO_NET_RESOURCE_OBJ_IPSEC_OUTB_SA	IPsec Outbound SA resource object
0x0204	VIRTIO_NET_RESOURCE_OBJ_IPSEC_INB_SA	IPsec Inbound SA resource object

```

struct virtio_crypto_resource_obj_ipsec_sa_256b_key {
    u8 mode;
    u8 direction;
    u8 reserved[2];
    le32 obj_id;
    le32 spi;
    le32 salt;
    le64 options;
    struct virtio_crypto_ipsec_tunnel_param param;
    le64 esn;
    le16 udp_sport;
    le16 udp_dport;
    le32 replay_win_sz;
    le64 cipher_algo;
    struct {
        le16 length;
        u8 data[32];
    } cipher_key;
    le64 auth_algo;
    struct {
        le16 length;
        u8 data[32];
    } auth_key;
}
  
```

Extended Sequence Number Support
 UDP encapsulation
 DSCP copy support
 DF copy support
 TTL decrement support



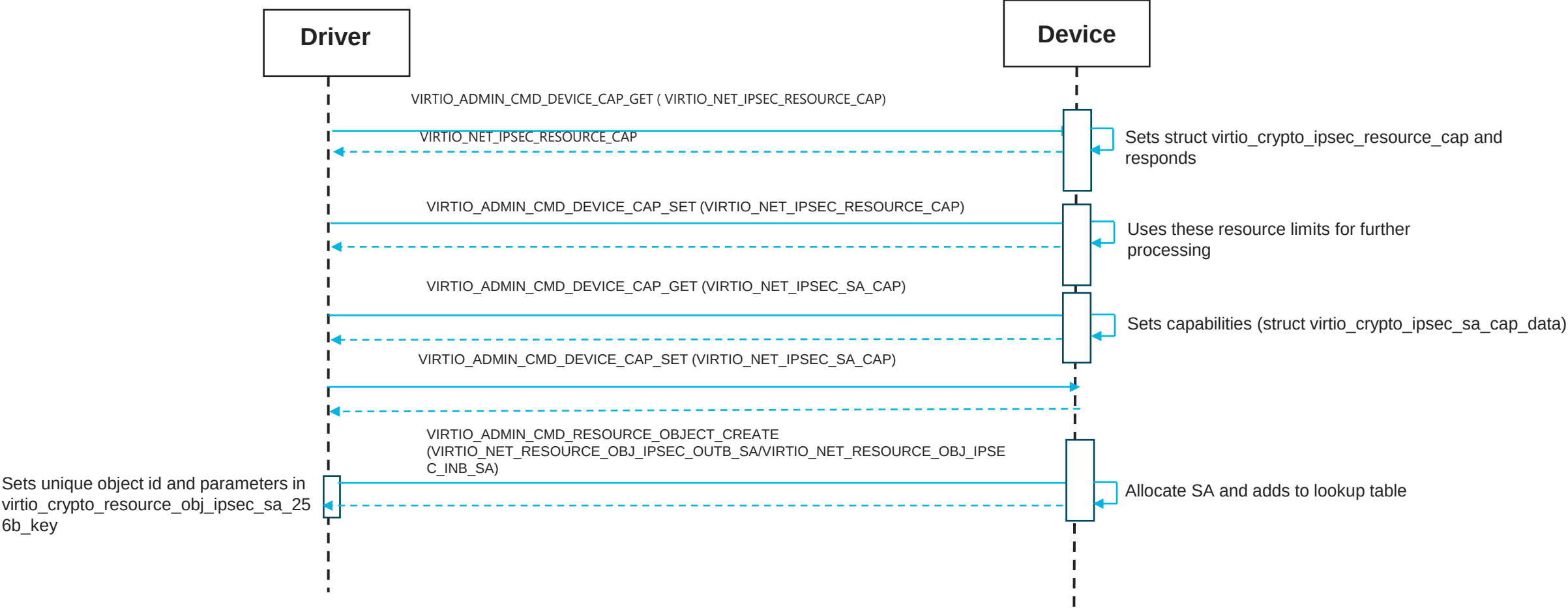
```

struct virtio_crypto_ipsec_tunnel_param {
    /* Tunnel type: IPv4 or IPv6 */
    u8 type;
    u8 reserved[3];
    union {
        /* IPv4 tunnel header parameters */
        struct {
            /* IPv4 source address */
            struct in_addr src_ip;
            /* IPv4 destination address */
            struct in_addr dst_ip;
            /* IPv4 Differentiated Services Code Point */
            u8 dscp;
            /* IPv4 Don't Fragment bit */
            u8 df;
            /* IPv4 Time To Live */
            u8 ttl;
        } ipv4;
        /* IPv6 tunnel header parameters */
        struct {
            /* IPv6 source address */
            struct in6_addr src_addr;
            /* IPv6 destination address */
            struct in6_addr dst_addr;
            /* IPv6 flow label */
            le32 flabel;
            /* IPv6 hop limit */
            u8 hlimit;
            /* IPv6 Differentiated Services Code Point */
            u8 dscp;
        } ipv6;
    };
};
  
```

Flow-Filter Actions

Action	Name	Description
0x3	VIRTIO_NET_FF_ACTION_IPSEC	Matching packet will undergo IPsec processing
0x4	VIRTIO_NET_FF_ACTION_IPSEC_RECIRCULATE	Matching packet will first undergo IPsec processing, followed by the flow filter rules again

Virtio-Net IPsec Operation Control Path



Virtio-Net IPsec Operation Data Path

- IPsec processing introduces additional changes to *virtio_net_hdr* includes,

```
struct virtio_net_hdr {  
    #define VIRTIO_NET_HDR_F_NEEDS_CSUM 1  
    #define VIRTIO_NET_HDR_F_DATA_VALID 2  
    #define VIRTIO_NET_HDR_F_RSC_INFO 4  
    #define VIRTIO_NET_HDR_F_SECURITY 8  
    #define VIRTIO_NET_HDR_F_SECURITY_ERR 16  
    u8 flags;  
    ....  
    le16 padding_reserved; (Only if VIRTIO_NET_F_HASH_REPORT negotiated)  
    /* Present only if VIRTIO_NET_F_IPSEC negotiated */  
    struct ipsec_resource_hdr {  
        le32 resource_id;  
        le16 resource_type;  
    }  
};
```

Virtio-Net IPsec Outbound Data Path

Driver Flow

- Sets `VIRTIO_NET_HDR_F_SECURITY` flag in ***virtio_net_hdr:flags***
- Sets ***virtio_net_hdr:resource_id*** to already created object id using `VIRTIO_NET_RESOURCE_OBJ_IPSEC_OUTB_SA`
- Sets ***virtio_net_hdr:resource_type*** to `VIRTIO_NET_RESOURCE_OBJ_IPSEC_OUTB_SA`

Device Flow

- Device performs the following if it finds `VIRTIO_NET_HDR_F_SECURITY` flag in ***virtio_net_hdr:flags***
 - SA lookup using ***virtio_net_hdr:resource_id***
 - Adds ESP/AH header
 - Adds ESP trailer if ESP mode
 - Encryption and authentication processing and adds ICV
 - Outer IP header if tunnel mode
 - Sends the packet out

Virtio-Net IPsec Inbound Data Path

Device Flow

- Does SA lookup
- Decryption and authentication processing using the SA parameters
- Removes tunnel header if in tunnel mode
- Removes ESP/AH header
- Remove the ESP trailer
- Sets the flag bit ***VIRTIO_NET_HDR_F_SECURITY / VIRTIO_NET_HDR_F_SECURITY_ERR*** in struct ***virtio_net_hdr:flags***
- Sets the ***virtio_net_hdr:resource_id***

Driver Flow

- Checks ***virtio_net_hdr->flags*** for ***VIRTIO_NET_HDR_F_SECURITY / VIRTIO_NET_HDR_F_SECURITY_ERR***
- Post process if any by retrieving the SA using the ***virtio_net_hdr:resource_id***

Summary

- Virtio-Net IPsec RFC Specification Status
 - <https://lore.kernel.org/virtio-comment/20250122062144.266725-1-schalla@marvell.com>
- Upstream vDPA driver for Marvell DPU processors
 - https://github.com/torvalds/linux/tree/master/drivers/vdpa/octeon_ep
- Future Work
 - Integrate Virtio-Net IPsec Operation into the Virtio Specification.
 - Support Inline IPsec in virtio-net driver for Marvell's DPU devices using Marvell's vDPA driver.
 - Introduce more IPsec options such as checksum computation or verification and tunnel header verification.

Q&A



Thank You