JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Status of Individual Drafts

IETF 126 DTN WG

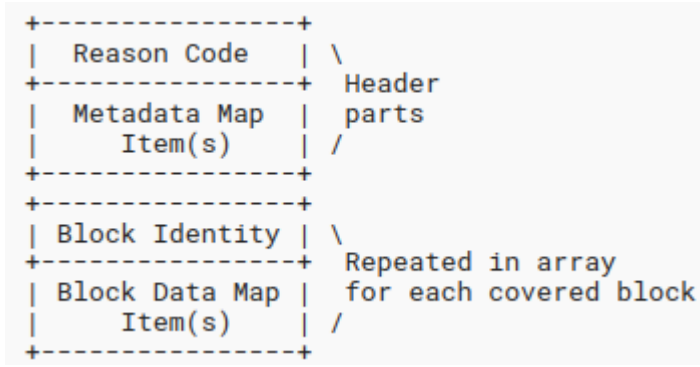
Brian Sipos
JHU/APL

Current Individual Drafts

- BP Manifest Block (draft-sipos-dtn-manifest-block-01)
- BP Security Associations with Few Exchanges (SAFE) (draft-sipos-dtn-bp-safe-01)
- BP Edge Node with Zero-Configuration (draft-sipos-dtn-edge-zeroconf-01)

BP Manifest Block

- This is a general purpose container for data derived from other, pre-existing blocks in the same bundle
- This block type supports a variety of use cases explained in the draft
- The manifest has been identified as a mechanism to support of Secure BP Audit Mechanism (SBAM)
- This block type definition is stand-alone and does not technically depend on any other drafts
- Changes from -00 to -01:
  - Reworked the encoded structure (similar to BP SAND) to avoid dependence on map deterministic encoding:
    1. Purpose code
    2. Metadata Map
    3. Array of:
       a. Covered block identity
       b. Block Data Map
  - New encoded structure makes inspecting/filtering reasons a quick activity
  - Added a Key ID item for security blocks with explanations about its creation and use




```
ext-data-manifest = [
  reason: int16,
  metadata-map,
  [* blockdata-grp]
]

; Tuple for each covered block
blockdata-grp = (
  ; From the IANA Bundle Block Types registry
  block-type: uint,
  ; Cross-reference to covered block
  block-num: uint,
  blockdata-map
)

; Controlled by metadata-item socket
metadata-map = {* $$metadata-item } .within manifest-structure
; Controlled by blockdata-item socket
blockdata-map = {* $$blockdata-item } .within manifest-structure

; General structure of all maps in the manifest
manifest-structure = {
  * int16 => any
}
int16 = -32768 .. 32767
```

BP Security Associations with Few Exchanges (SAFE)

- The DTN WG charter includes topics of “OAM and Key Management”
- Key management includes parallel topics of:
  - Key **distribution** from a centralized entity (which cannot provide forward secrecy by NIST definition)
  - **Credential** management (credentials bind identity and other parameters to a public key)
  - Key **derivation** or **delegation** of authority (i.e. public key certificate hierarchy)
  - Key **agreement** between autonomous peers (which can involve forward secrecy)
  - Key upkeep (i.e. revocation, re-key/roll-over)
- This BP SAFE individual draft provides a starting point for discussion
- The initialization using Ephemeral Diffie-Helman Over COSE (EDHOC) and primary SA agreement are in a good state
- The ability to pre-position “public pre-keys” and perform “unilateral rekey” is a unique feature to this protocol (I think)
  - This enables delay-independent rekey while providing forward secrecy to exchanged session keys
- There have been no recent updates to this document
- A design summary is in a few following slides

BP SAFE Use Cases

- Four use cases currently identified, with different levels of detail currently defined.
- **End-to-end security**: source at the application transmit and acceptor at the application delivery
  - Covers target blocks for the entire bundle lifetime
- **One-hop transport security**: source at the forwarder acceptor at the immediate receiver
  - Supplements underlayer-provided transport security
  - Allows uniformity of security guarantees (i.e. Previous Node authentication)
  - Has operational limitations regarding target lifecycle
  - Depends on BPA supplying next-hop info to BPSec entity before forwarding
- **Multi-hop transport security**: most general-purpose and complex between any forwarder and any receiver
  - This use case requires not just security configuration but routing control (either knowledge or forcing)
- **Tunnel security**: is still TBD but useful to explain
  - Intended to relate to BIBE tunnelling of BPv7
  - Even if not supported by BP SAFE, can be explicitly carved out

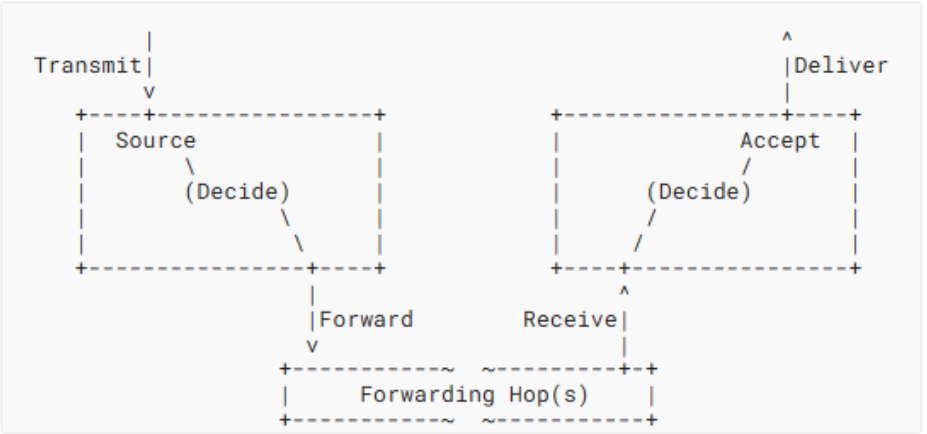


Figure 3: End-to-End security flows

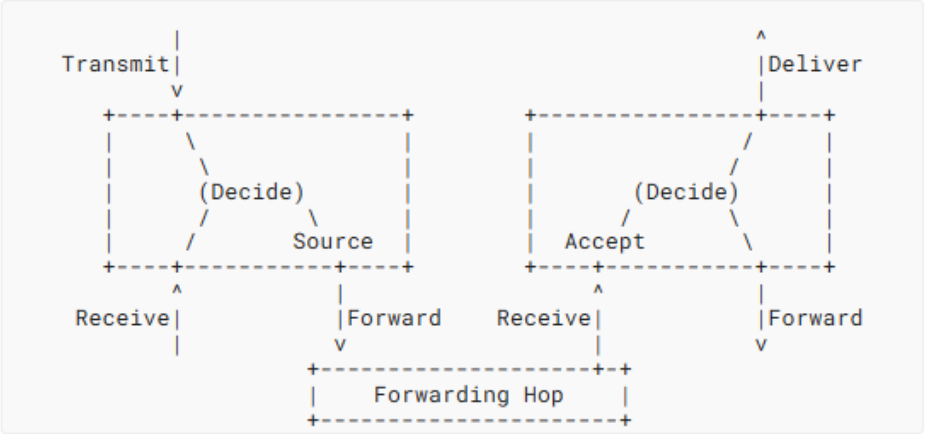


Figure 4: One-Hop security flows

BP SAFE Information Bases

- These logical structures ensure protocol entities manage the needed set of information with associated constraints
- **Participating Peers**: who is authorized, can be configuration
- **Activity States**: simple sequence bookkeeping
- **Primary Security Associations** (PSA):
  - Each peer has a single PSA
  - Enables SAFE message security
  - Each side of the SA can re-key unilaterally
  - Each rekey has optional forward secrecy
- **Secondary Security Associations** (SSA):
  - Each BPSec-targeted flow (at a block-granularity) has a single SSA
  - Derived key material needs are driven by a specific BPSec security context associated with each SSA

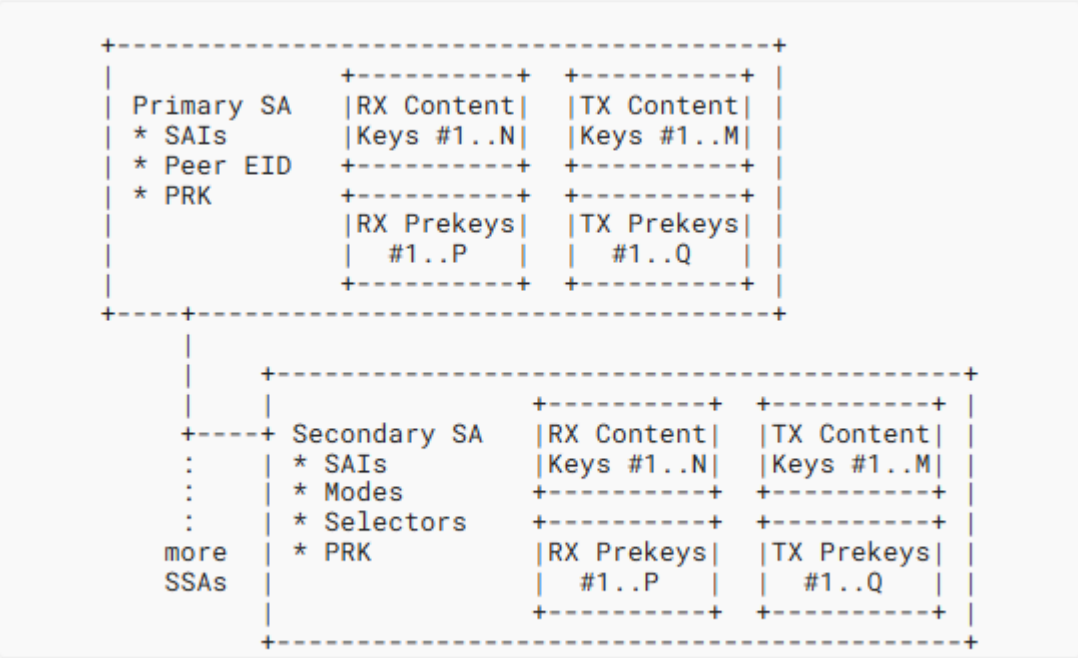


Figure 5: Relationships between SAs and their key material

BP SAFE Message Structure

- A clear separation of sub-layers makes each have a specific narrow function with simple logic
- This structure separates common logic of **activity sequencing** and selective retransmission from the higher-level purpose of each activity
- This structure allows multiple independent, byte-string-wrapped messages to be transported in each SAFE PDU
- Authenticated **confidentiality** (AEAD) at the SAFE layer ensures that its correct operation occurs independently of whatever BPSec context support exists in the underlying BPA on each node
  - Guarantees minimum interoperability

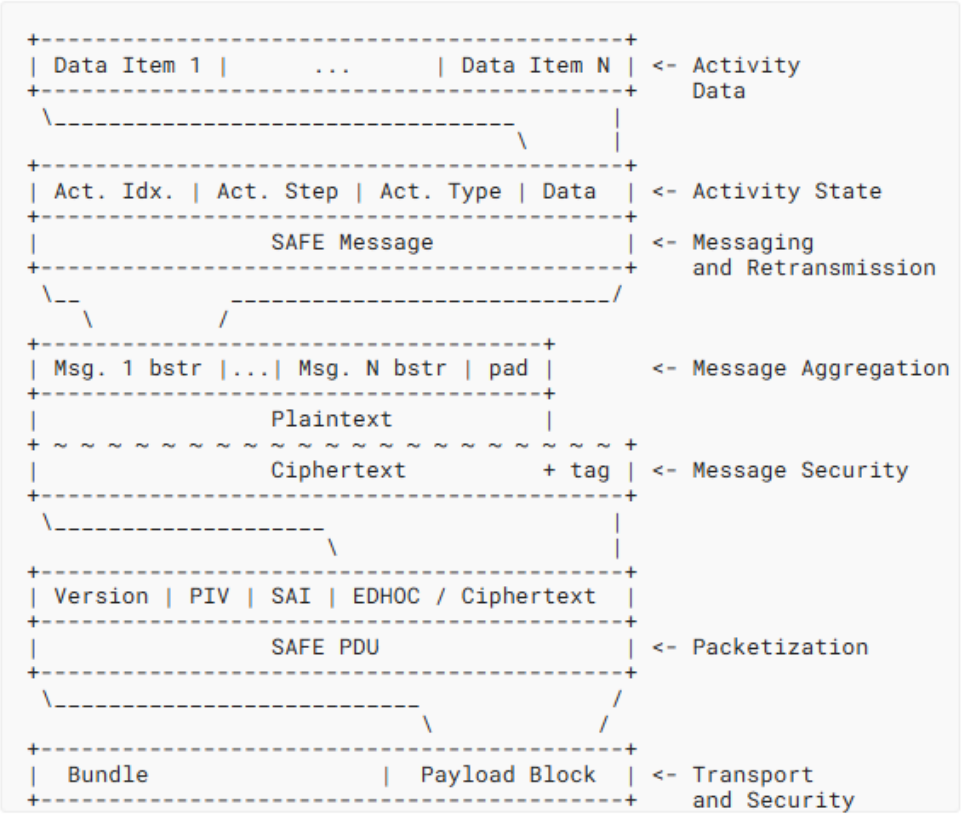
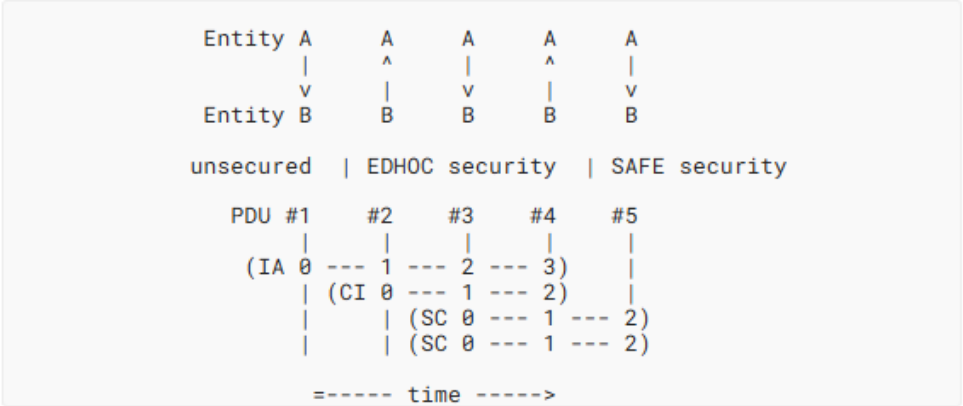
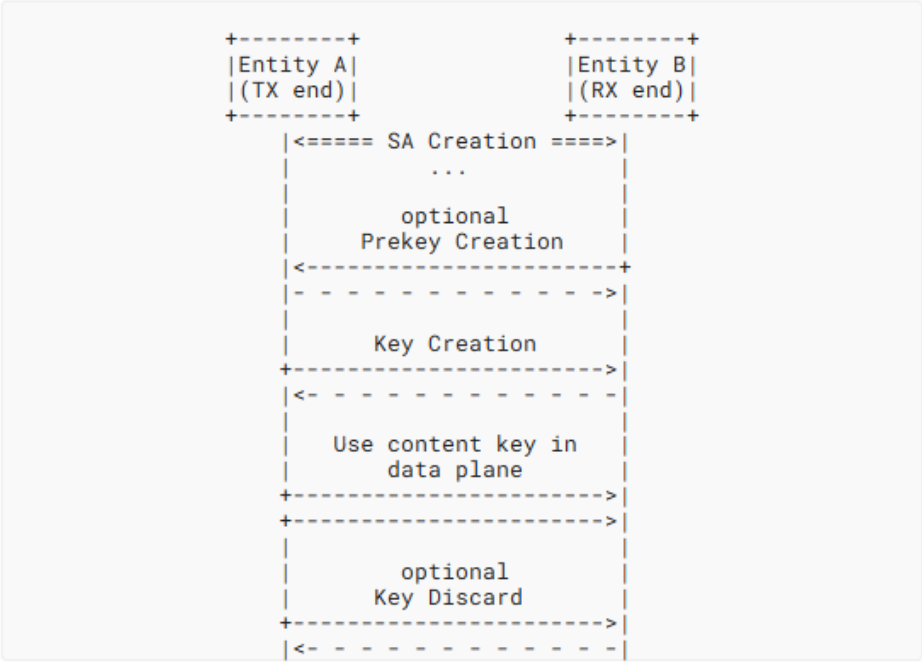


Figure 8: Breakdown of sub-layers within SAFE

BP SAFE Activity Sequencing

- Each **activity** is a short sequence of message exchanges between two peers
- Activity processing is always lock-step between peers, acknowledging with selective retransmission
- Some activity types involve negotiation between peers, meaning:
  - The **initiator** of the activity proposes acceptable options
  - The **responder** chooses (and narrows) one option
- Many activity types for upkeep are few or even one-step (but with acknowledgement)
- Each activity step is communicated by a separate SAFE message
- Messages are combined in PDUs to allow pipelining of activities between peers





BP SAFE Ongoing and Next Work

- Need implementation experience for specific secondary SA use cases, specifically:
  - Options for flow selectors (down to block-specificity for BPSec targets)
  - Options for each supported security context
- Work in IETF LAKE WG is ongoing to enable post-quantum crypto (PQC) for EDHOC
  - One PQC aspect: initial session confidentiality via key encapsulation mechanism (KEM)
  - Another aspect: authentication via signature or KEM private key proof-of-possession
  - We will be able to use the results of this “for free”
- A number of open issues are being tracked in the SAFE source repository

BP Edge Node with Zero-Configuration

- Goal is to allow using a BPA in a stable IP LAN (IPv4, IPv6, mixed) without spending time configuring nodes
  - Allow BP application demos to focus on *using the apps* not on configuring the network
- This is a purely informational document for how to use existing mDNS and DNS service discovery
- Could be updated to include additional example of using existing UDPCL service name
- A proof of concept was implemented in the dtn-demo-agent
- There have been no new updates to this document