Architecture for Delay-Tolerant Key Administration

IETF100 DTN Working Group
November 16, 2017

Scott Burleigh (Scott.Burleigh@jpl.nasa.gov)

David Horres (David.C.Horres@jpl.nasa.gov)

Kapali Viswanathan (kapaleeswaran.viswanathan@boeing.com)

Michael W. Benson (michael.w.benson@boeing.com)

Fred L. Templin (fred.l.templin@boeing.com)

Motivation

- On-demand & interactive communication cannot be assumed in DTN
- SSL and Online Certificate Status Protocol (OCSP) require on-demand & interactive communication
- A DTN-friendly public-key distribution and revocation protocol suite is needed

Delay-Tolerant Key Administration (DTKA)

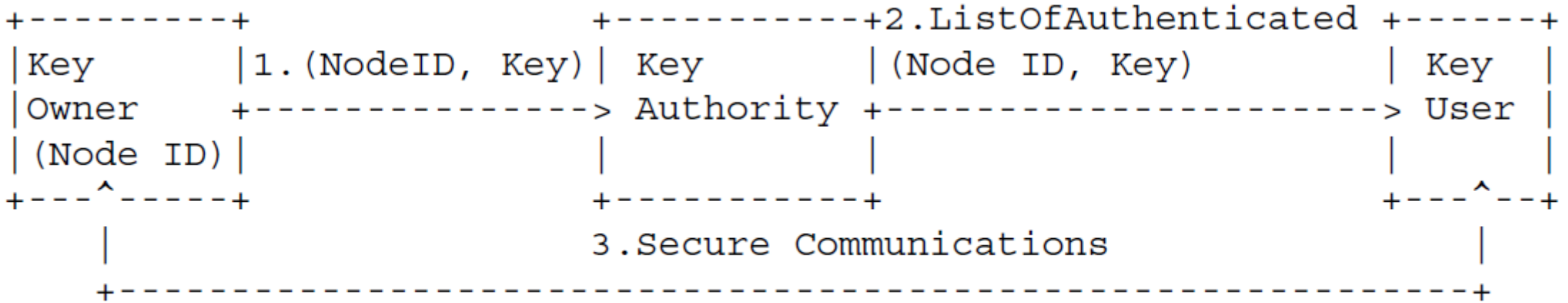


Figure 1: Abstract Data-Flow-Diagram for DTKA

- Key Owner: has private key & claims ownership of corresponding public key
- Key Authority: verifies ownership of public key & authenticates ownership of public key for Key Owner
- Key User: uses public key authentication by Key Authority to securely communicate with Key Owner

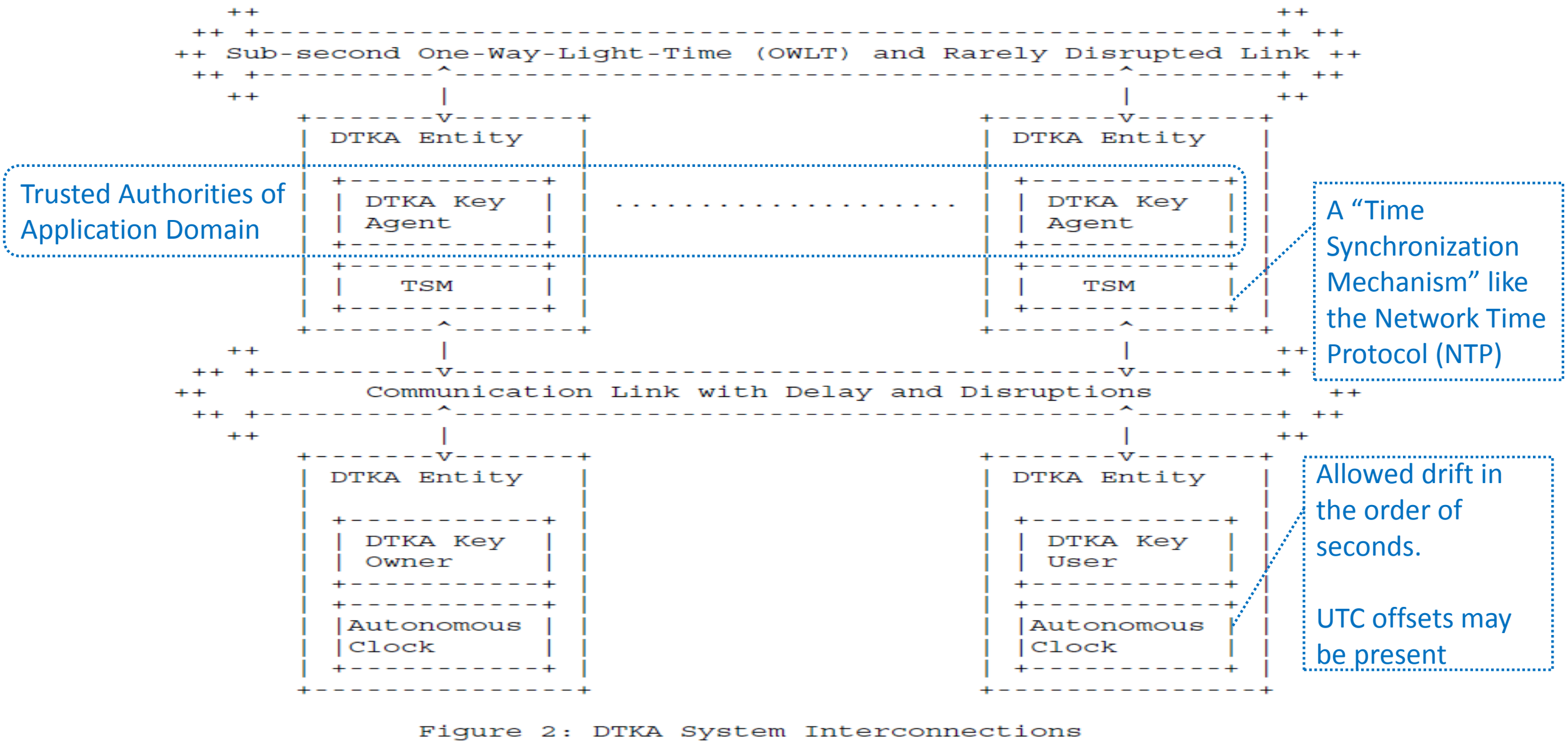


Figure 2: DTKA System Interconnections

System Security Configuration:
Public key of each DTKA Key Agent is securely configured into every Agent, Owner and User in the application domain

Avoiding single-point-of-trust

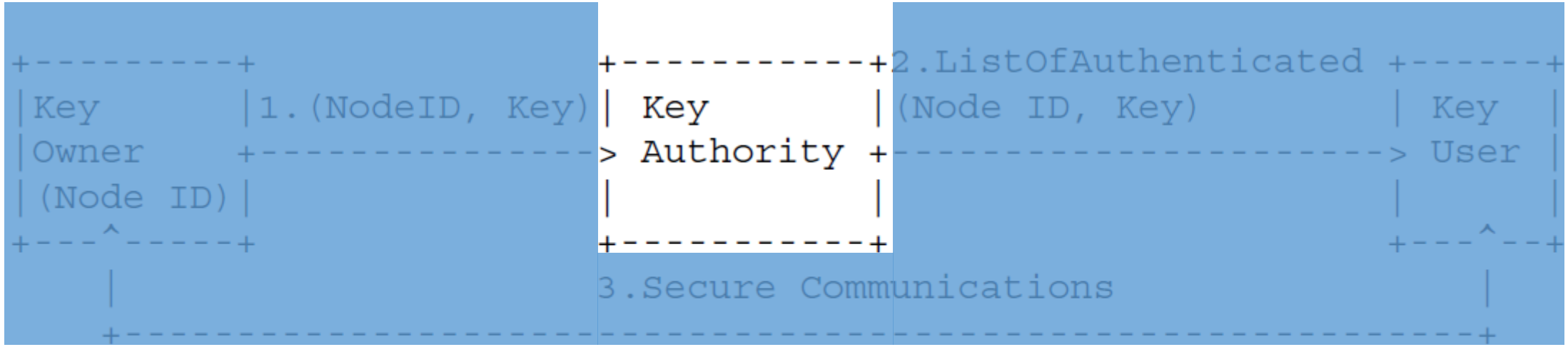


Figure 1: Abstract Data-Flow-Diagram for DTKA

- Role of Key Authority is distributed among a set of Key Agents
  - To prevent single-points of failure
  - i.e. A pre-defined set of Key Agents constitute the Key Authority
- More on this topic later in the discussion

Node/key registration

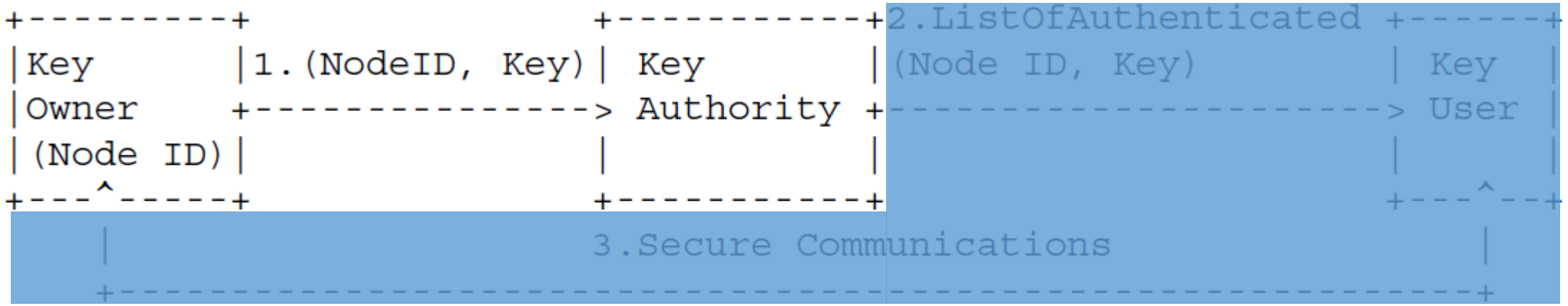
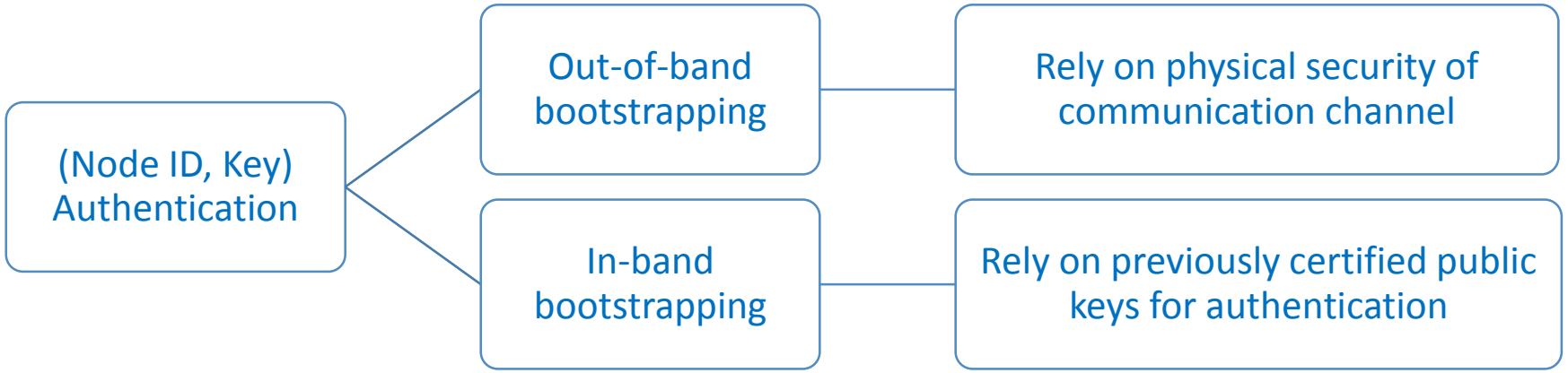


Figure 1: Abstract Data-Flow-Diagram for DTKA



https://tools.ietf.org/html/draft-burleigh-dtnwg-dtka-00

Periodic key updates

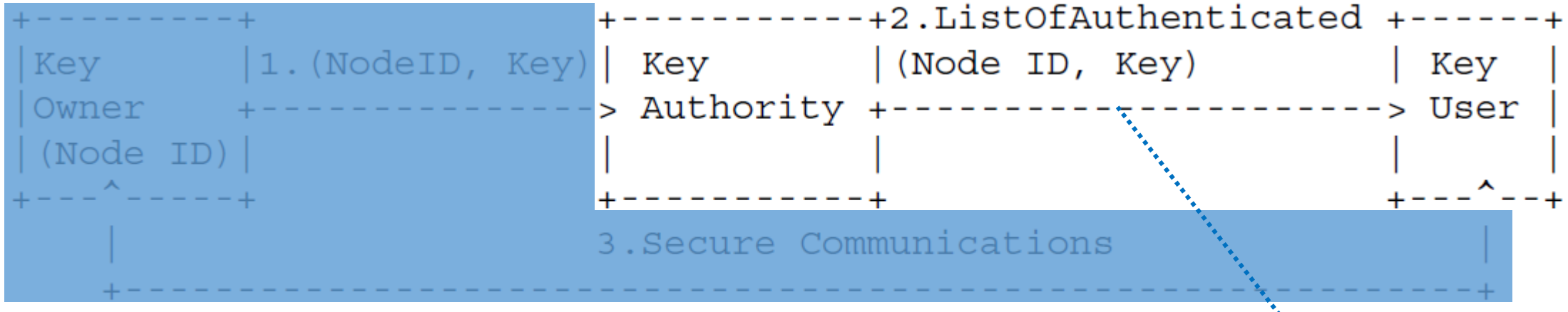


Figure 1: Abstract Data-Flow-Diagram for DTKA

Communication protected using Bundle Integrity Block of DTN employing the Key Authority's private key

- Key Authority dispatches periodic updates to list of authenticated keys
- Instantaneous receipt of periodic key updates is not assumed

Message Format for Key Updates

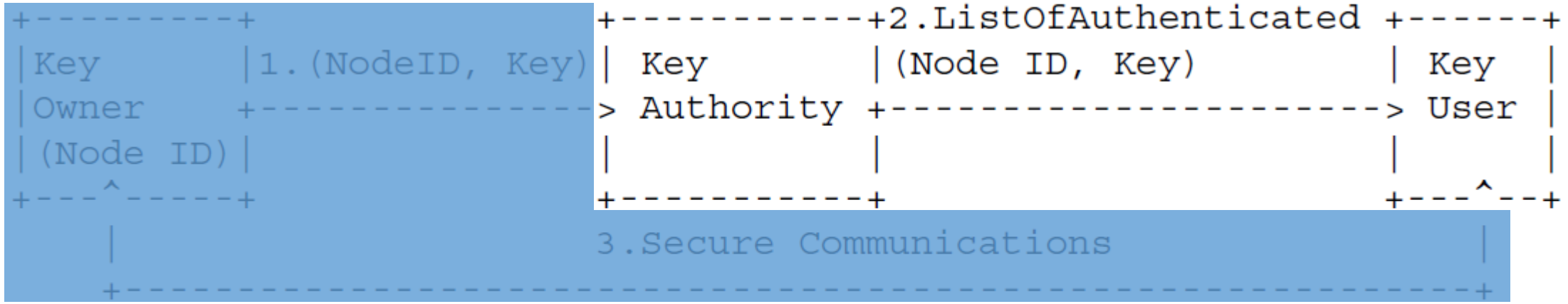


Figure 1: Abstract Data-Flow-Diagram for DTKA

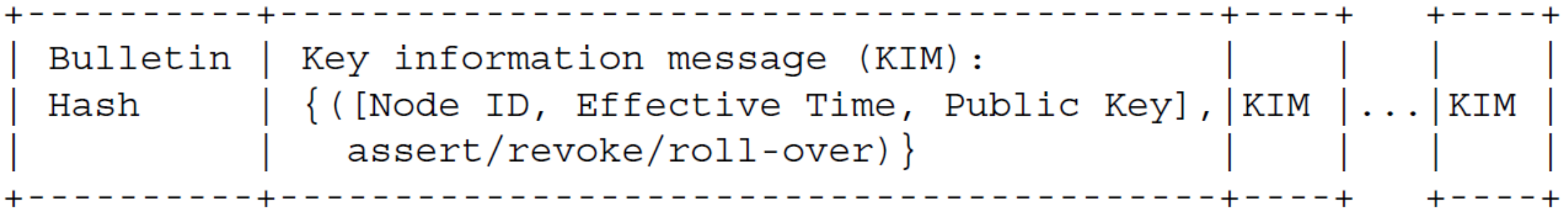


Figure 3: Bulletin

Code blocks formed by each Key Authority

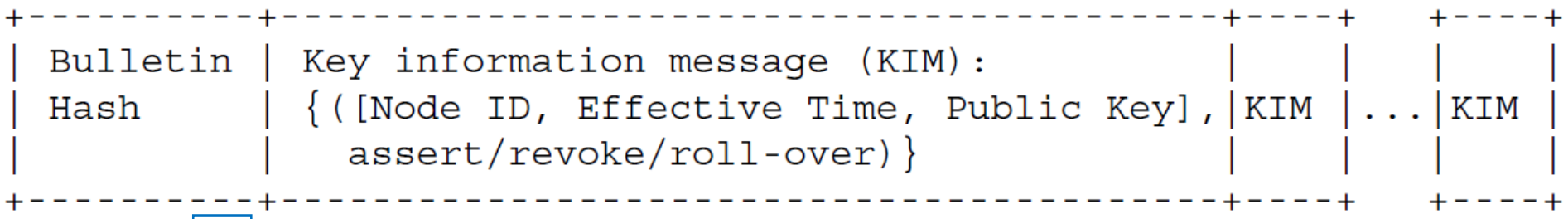


Figure 3: Bulletin

Bulletin formation using (Q+k) erasure coding @ Key Authority X

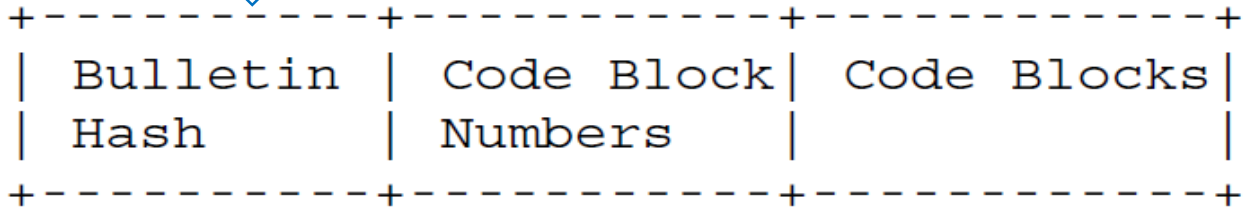


Figure 4: Message Format for Code Blocks

https://tools.ietf.org/html/draft-burleigh-dtnwg-dtka-00

Code block assignment for Key Authorities

Code Block Numbers (0 to (Q + k -1))	0	1	2	3	4	5	6	7
KA 1	x	x	x					
KA 2		x	x	x				
KA 3			x	x	x			
KA 4				x	x	x		
KA 5					x	x	x	
KA 6						x	x	x
KA 7	x						x	x
KA 8	x	x						x

Table 1: Example: Code Blocks Assignments for Key Authorities

Security configuration: All DTKA entities are configured to accept only the designated code blocks from the key agents. Eg: Code block 7 is accepted only from KAs 6, 7, and 8

- Scheme
  - A (Q=7, k=1) erasure code is used
  - 5-out-of-8 (t-out-of-n) Key Authorities needs to be received bulletin reconstruction

Summary

- Delay-Tolerant Key Administration (DTKA) is a public-key distribution and revocation mechanism for DTN
- Uses Bundle Protocol Security's Bundle Integrity Block (BIB) to provide authentication
- Uses (Q,k) erasure coding to distribute the trust in Key Authority among multiple Key Agents
  - Realizes the no-single point of failure requirement for DTKA
- Internet Draft has detailed design for the DTKA protocol suite

Thank you!

Additional slides

Node/key registration

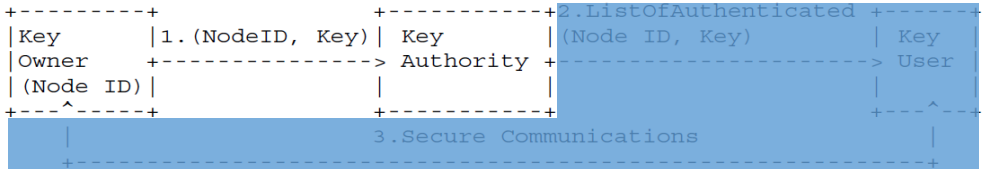


Figure 1: Abstract Data-Flow-Diagram for DTKA

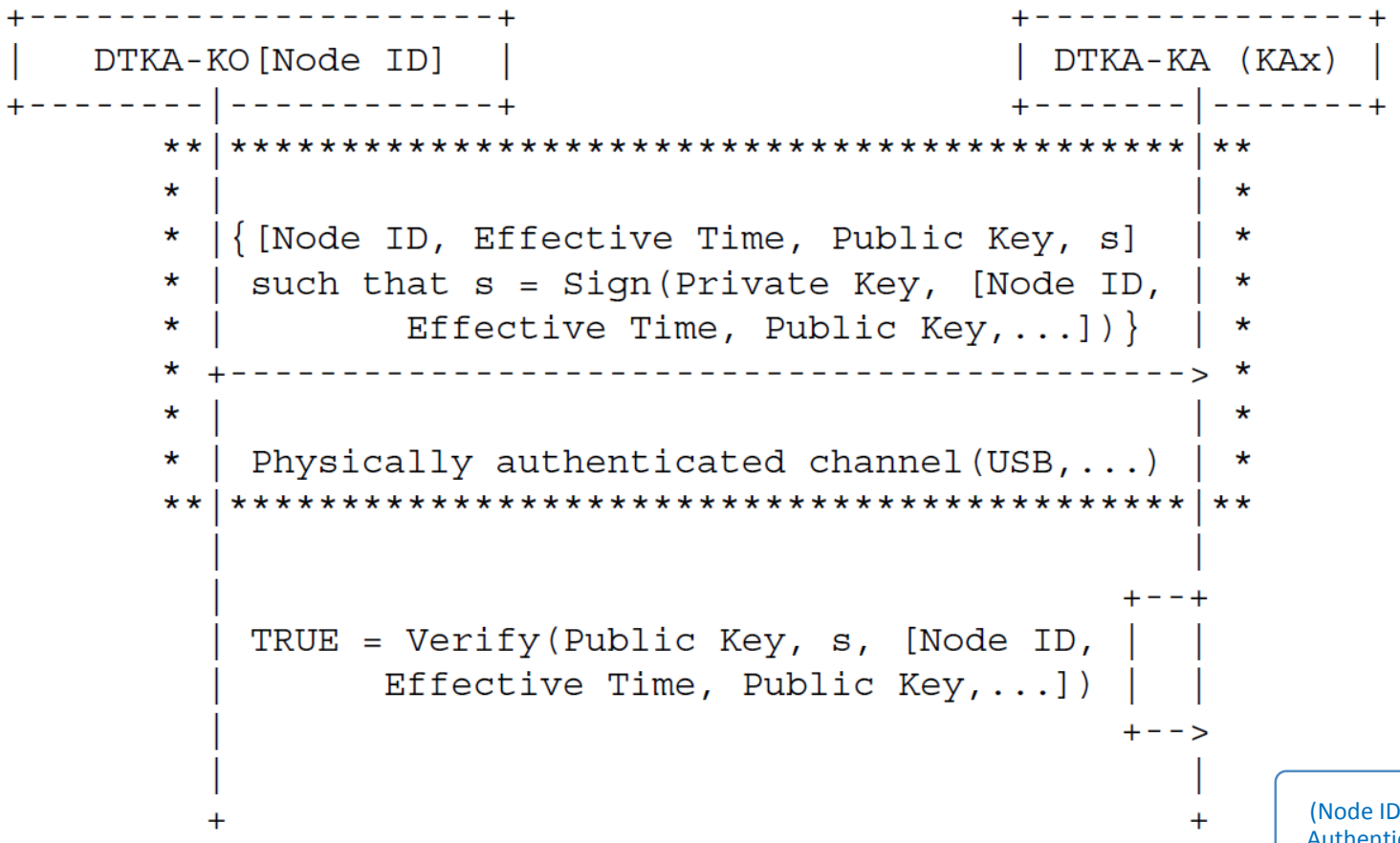
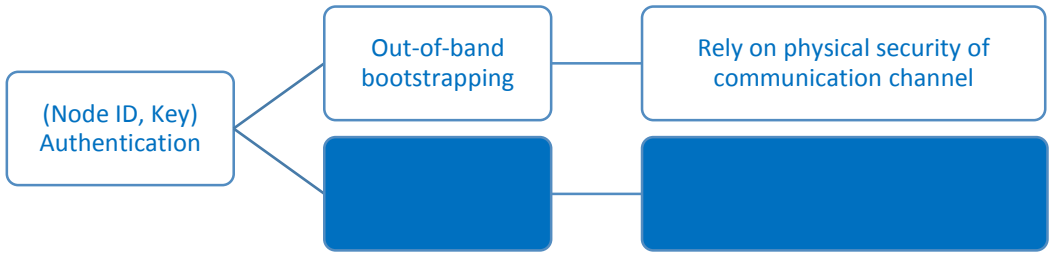


Figure 5: Interaction Diagram 1: Node Registration



Node/key revocation

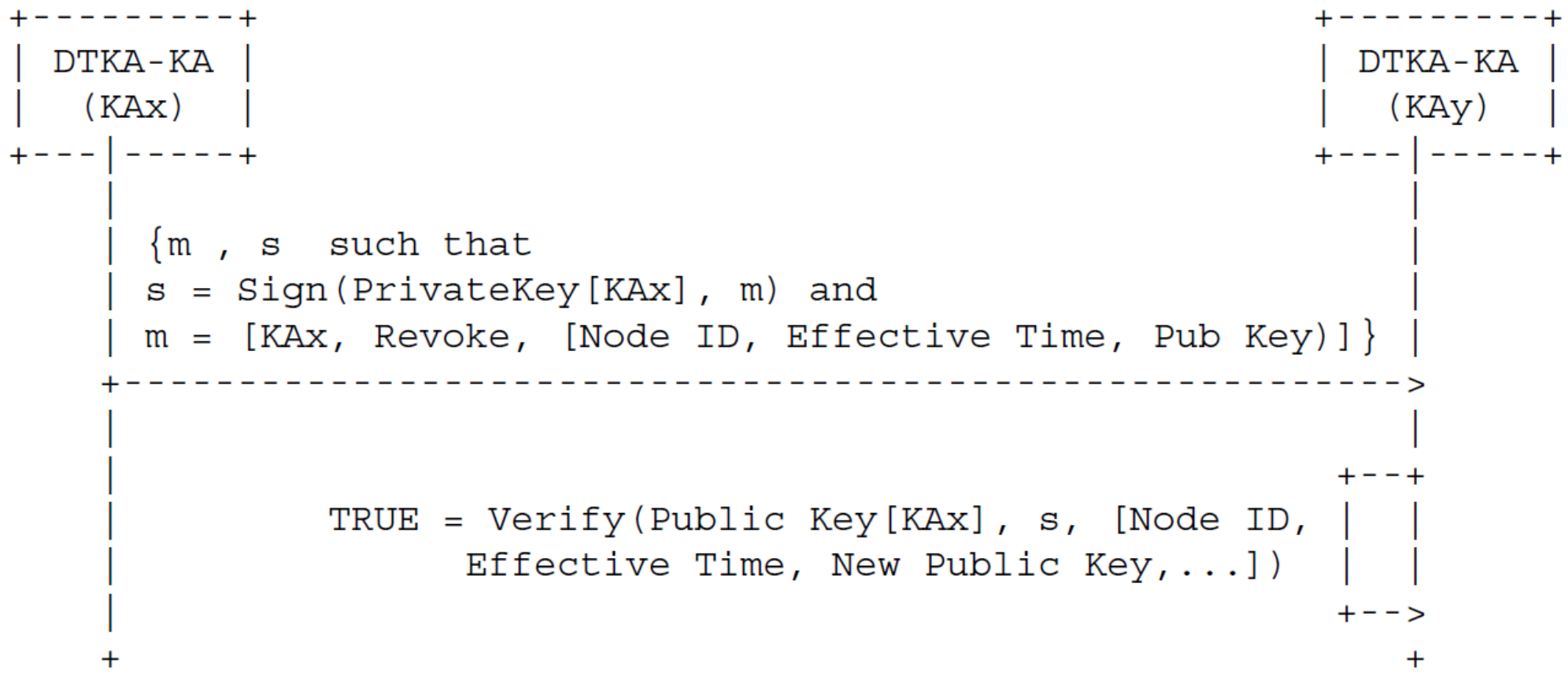


Figure 6: Interaction Diagram 1.1: Key Revocation

Node/key rollover

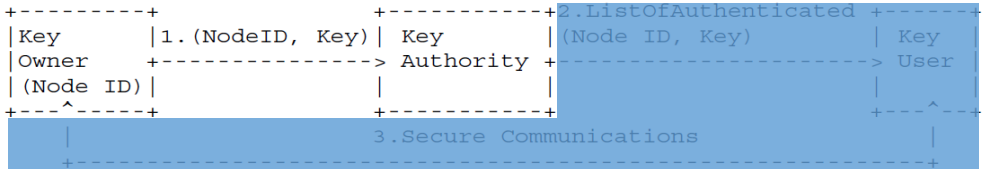


Figure 1: Abstract Data-Flow-Diagram for DTKA

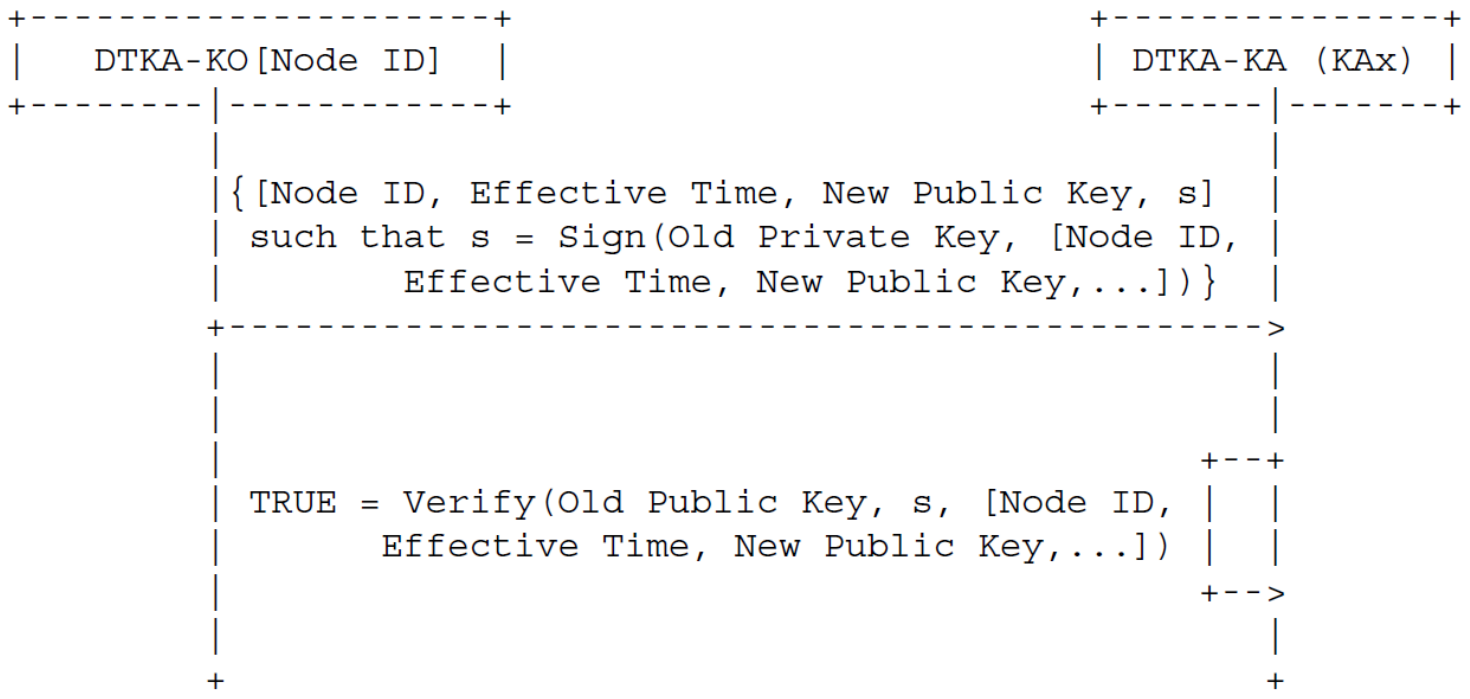
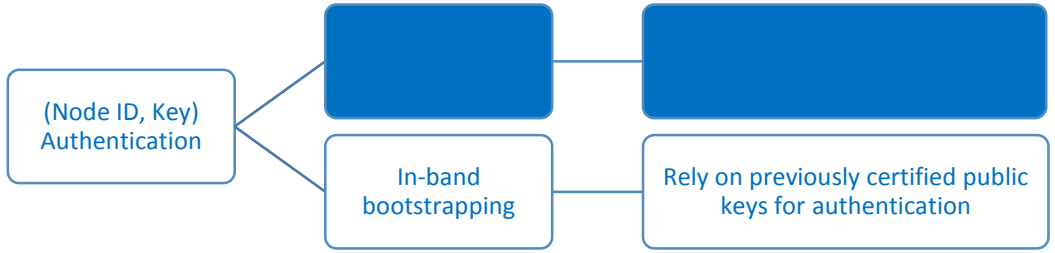


Figure 7: Interaction Diagram 1.2: Key Rollover



Key distribution

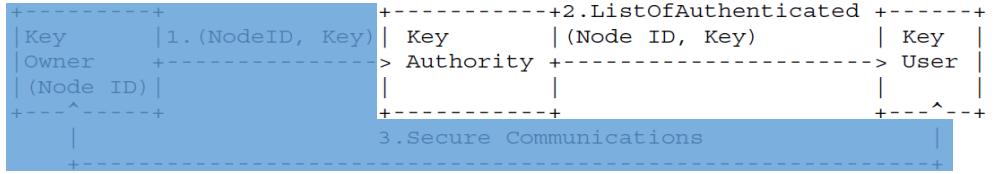


Figure 1: Abstract Data-Flow-Diagram for DTKA

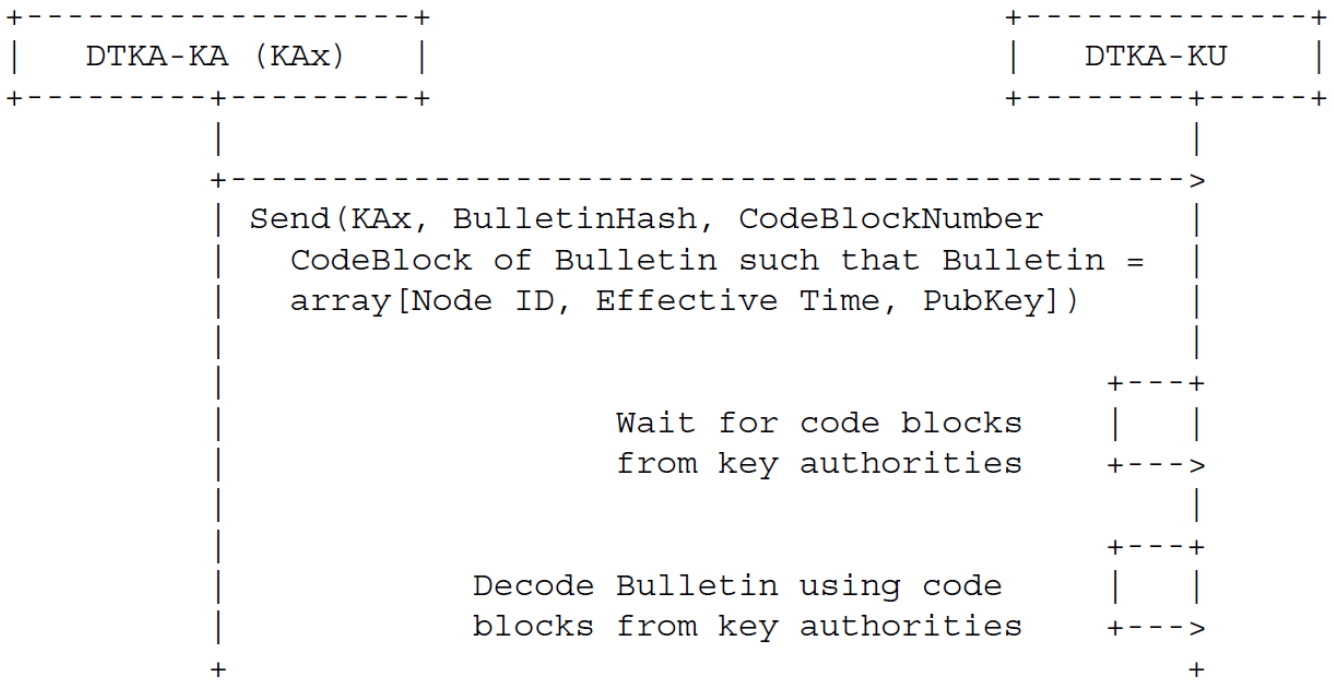


Figure 8: Interaction Diagram 2: Bulk Key Distribution

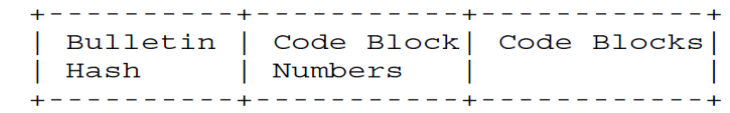


Figure 4: Message Format for Code Blocks

(Q+k) erasure decoding with DTKA rules for accepting code blocks

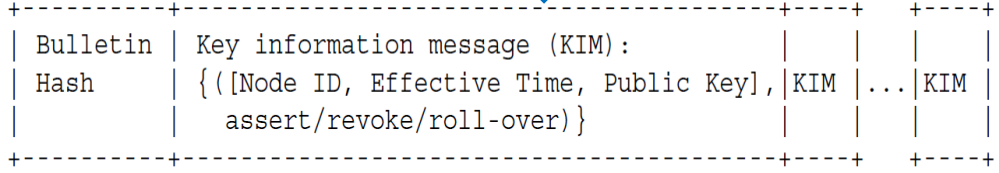


Figure 3: Bulletin