Integrity of In-situ OAM Data Fields

draft-ietf-ippm-ioam-data-integrity-07

Justin Iurman, Frank Brockners, Shwetha Bhandari, Tal Mizrahi

IETF 119, IPPM WG
March 19, 2024

Status -07

- Submitted before secdir review
- Working on next version…
- Challenge: different possibilities, lots of compromises
- Looking for WG feedback

Secdir review: DISCUSS points

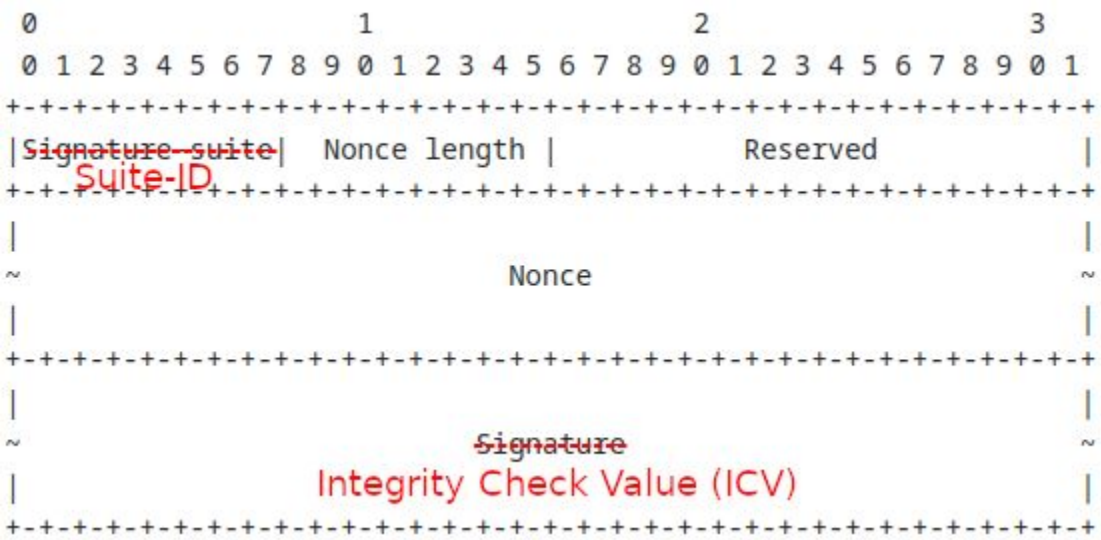
Solved DISCUSS (editorial changes):

- Signature vs (G)MAC
- GCM Key usage limitations
- Nonce guidance
- “Signature” as nonce for transit nodes

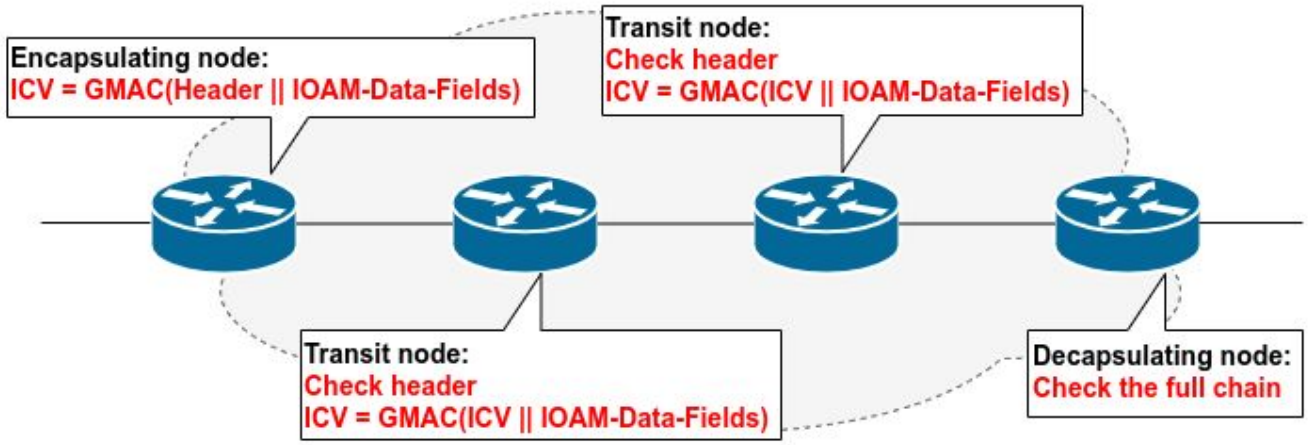
Pending DISCUSS (depends on the chosen option, see next slides):

- Header fields selection for integrity protection

Reminder: IOAM Integrity Protection Header

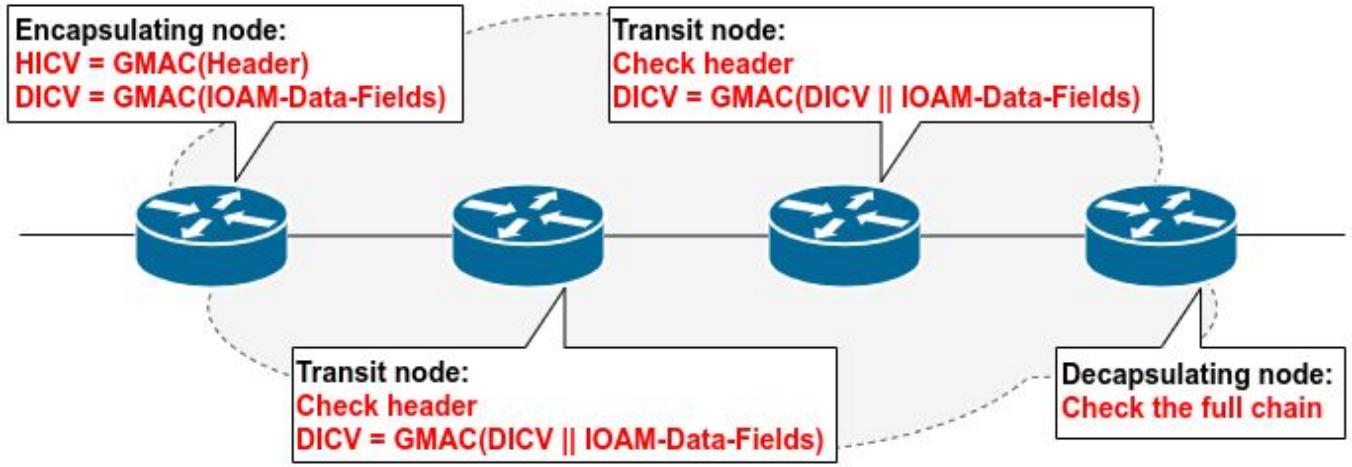


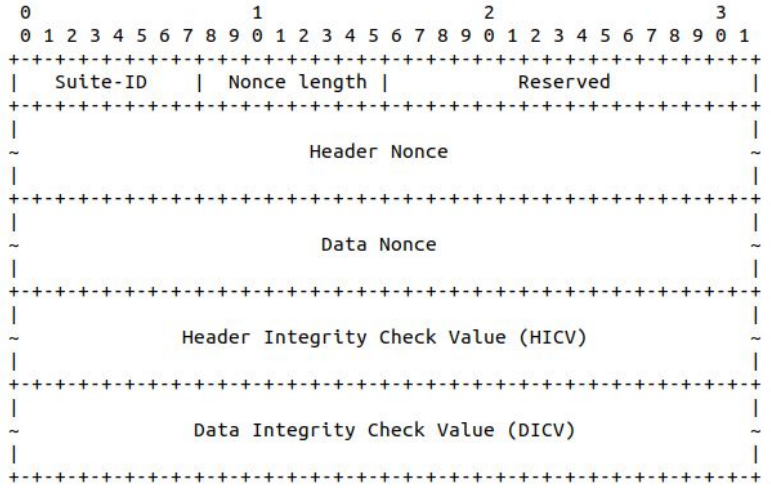
Option 1a: Validation at the end (w/ header check)



- Currently: If a transit node processes a field/flag triggering actions from the node, then the node MUST check the header (e.g., DEX, Trace Loopback flag)
- A transit node checks the header by recomputing ICVs from nodes 0 to n-1 (and so for each transit node!!!)
- Each IOAM node requires the keys from all prior nodes
- Pending DISCUSS: **unsolved**

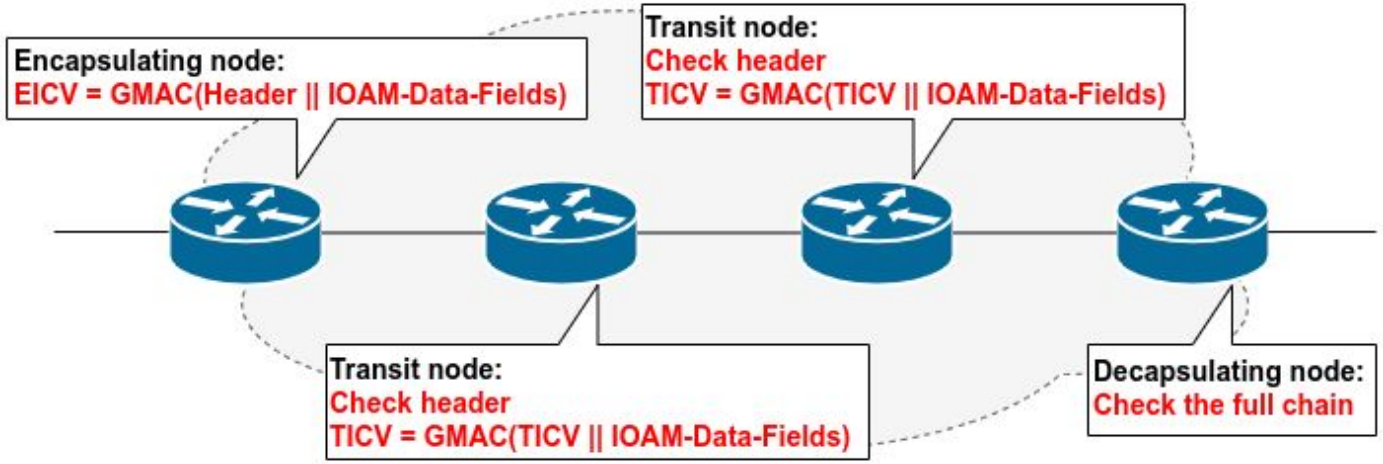
Option 1b: Validation at the end (w/ header check)

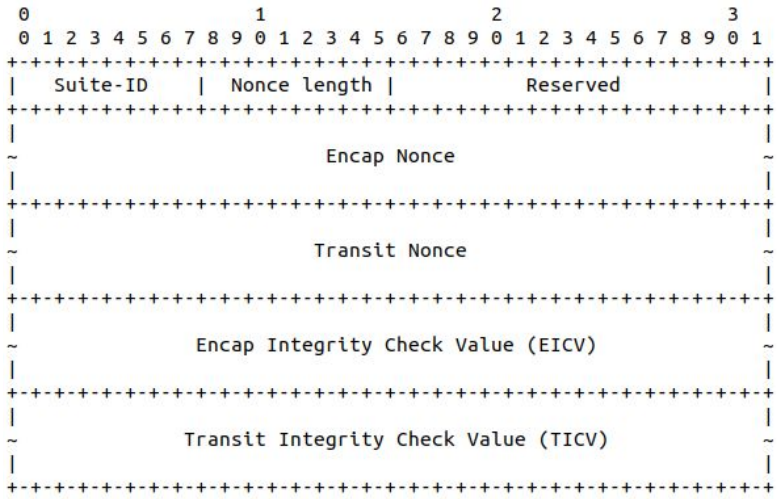




- Extra ICV, one-step header verification for each transit node (can now be applied all the time)
- Encapsulating node performs GMAC 2 times (i.e., one for the header and the other one for IOAM-Data-Fields)
- Each IOAM node requires the key from the encapsulating node
- Pending DISCUSS: **unsolved**

Option 1c: Validation at the end (w/ header check)





- Change ICV semantics: the encapsulating node performs only one GMAC, and it's still a one-step header verification for each transit node
- Transit nodes need to fetch and include the encapsulating node's IOAM-Data-Fields to check the header (worst case: when the Opaque State Snapshot is required → must parse the entire trace from top to bottom)
- Each IOAM node requires the key from the encapsulating node
- Pending DISCUSS: **unsolved**

Considerations on 1a, 1b, 1c

Common problem between 1a, 1b and 1c?

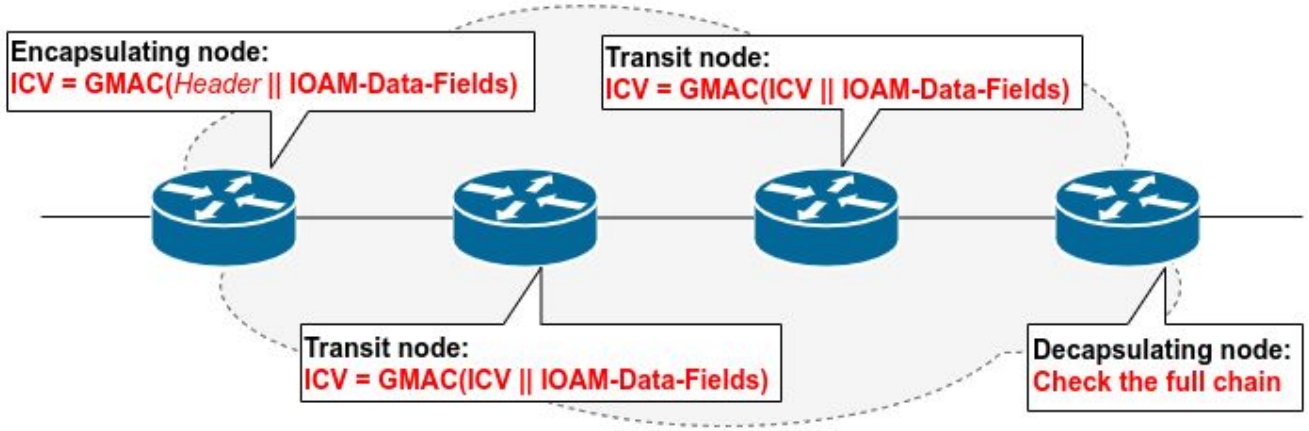
- header check = *IOAM nodes receive the key of the encapsulating node*
- we have to trust all IOAM nodes (i.e., **not a full integrity protection**)

Alternative solution: **no header check**.

→ Focus on original objective of IOAM integrity protection:

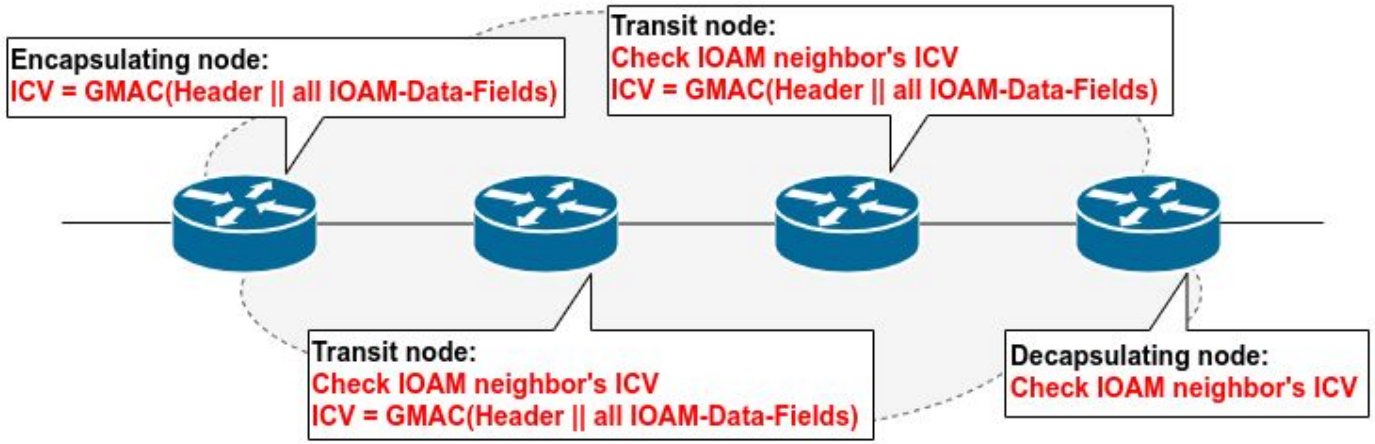
- protection of **IOAM-Data-Fields** (rather than the header)
- distinguish between header fields required for processing and header fields required for the interpretation of IOAM-Data-Fields (only the integrity protection of the latter is needed, e.g., Namespace-ID)
- triggering fields/flags out of scope, i.e., processing rather than integrity related (e.g., Loopback flag)

Option 2: Validation at the end (**no** header check)



- Faster processing on transit nodes, i.e., no header check
- The encapsulating node can include immutable header fields which are required for the interpretation of IOAM-Data-Fields, like e.g., the Namespace-ID
- Each IOAM node only shares its key with the Validator (= “don’t trust any node”)
- Pending DISCUSS: **solved**

Option 3: Neighbor validation



- Hop by hop validation (in this case, the entire header and all IOAM-Data-Fields)
- Requires that IOAM nodes are trusted
- Requires key distribution between all IOAM nodes
- Pending DISCUSS: **solved**

Option 4: IPSec

- Quite similar to solution 3, but does not require any new protocol
- IPSec tunnels configured between all IOAM nodes that match the physical topology/connectivity (all traffic with IOAM runs across the IPSec tunnels)
- Requires that IOAM nodes are trusted
- Pending DISCUSS: **solved**

Best solution?

n = number of nodes (from 1 to n)
i = node position (from 1 to n)

	Integrity Header	Header Check	Header Freeze	Full Protec.	Encap GMAC	Transit GMAC	Decap GMAC
Option 1a	1 ICV	~Yes	Yes	No	1	i	n-1
Option 1b	2 ICVs	Yes	Yes	No	2	2	n
Option 1c	2 ICVs	Yes	Yes	No	1	2	n-1
Option 2	1 ICV	~None	~No	Yes	1	1	n-1
Option 3	1 ICV	Yes	No	No	1	2	1
Option 4	AH/ESP	Yes	No	No	1	2	1

Protection of IOAM-Data-Fields as main objective

n = number of nodes (from 1 to n)
i = node position (from 1 to n)

	Integrity Header	Header Check	Header Freeze	Full Protec.	Encap GMAC	Transit GMAC	Decap GMAC
Option 1a	1 ICV	~Yes	Yes	No	1	i	n-1
Option 1b	2 ICVs	Yes	Yes	No	2	2	n
Option 1c	2 ICVs	Yes	Yes	No	1	2	n-1
Option 2	1 ICV	~None	~No	Yes	1	1	n-1
Option 3	1 ICV	Yes	No	No	1	2	1
Option 4	AH/ESP	Yes	No	No	1	2	1

Protect against person-in-the-middle attacks

n = number of nodes (from 1 to n)
i = node position (from 1 to n)

	Integrity Header	Header Check	Header Freeze	Full Protec.	Encap GMAC	Transit GMAC	Decap GMAC
Option 1a	1 ICV	~Yes	Yes	No	1	i	n-1
Option 1b	2 ICVs	Yes	Yes	No	2	2	n
Option 1c	2 ICVs	Yes	Yes	No	1	2	n-1
Option 2	1 ICV	~None	~No	Yes	1	1	n-1
Option 3	1 ICV	Yes	No	No	1	2	1
Option 4	AH/ESP	Yes	No	No	1	2	1

Proposal

- Focus the draft on integrity protection for IOAM-Data-Fields, i.e., “Option 2”
- Include a section in the draft that discusses the use of IPSec for deployments that are concerned about person-in-the-middle attacks, i.e., “Option 4”