

A Profile for Traffic Origin Authorizations (TOAs)

Authors: **Lancheng Qin** (Zhongguancun Lab)

Ben Maddison (Workonline)

Dan Li (Tsinghua University)

Igor Lubashev (Akamai)

Mar, 2026

Terminology

□ Route Origin of A Prefix

- ◆ The AS authorized to originate routes to the prefix in BGP

□ Traffic Origin of A Prefix

- ◆ The AS that originates traffic using the prefix as source addresses
- ◆ Generally, the set of traffic origins includes all route origins and may also contain additional entities that generate traffic but are not authorized to originate the route

Background

□ Key Challenge in SAV: Divergence Between Traffic Origin and Route Origin

- ◆ Current infrastructure for validating source addresses is mostly built on route-origin validation mechanisms
 - Typically uses **BGP UPDATE** or **RPKI ROA** (which represent destination information) to identify ASes authorized to source traffic from a prefix
- ◆ In most cases, works well when traffic origin = route origin
- ◆ **Issues arise when traffic origin \neq route origin**, causing legitimate traffic to be improperly blocked

traffic origin = route origin



Works well

traffic origin \neq route origin



Improper Block

Background

□ Why traffic origin \neq route origin happens

◆ It happens when route to the prefix is

- **not authorized to be announced by any AS (e.g., scenario #1), or**
- **authorized to be announced by another AS B, not the traffic origin AS A (e.g., scenario #2)**

Scenario #1

Unidirectional traffic that does not attract return traffic via route announcements

- Examples: IP multicast, Control/operational traffic using IANA special-purpose prefixes

Scenario #2

The return traffic is intentionally directed to the route origin, not the traffic origin

- Examples: CDNs using Direct Server Return (DSR)

Quick Review of TOA

□ Solution: TOA (Traffic Origin Authorization)

- ◆ A SAV-specific RPKI Signed Object which **authorizes an AS to source traffic from a given prefix (particularly when that AS is not authorized/intended to originate any route to that prefix)**
- ◆ Help SAV work when traffic origin \neq route origin
- ◆ Recommendation:
 - Operators SHOULD NOT register a TOA that is identical to or covered by an existing ROA

□ Applications/Compatibility

- ◆ TOAs can be easily used by several ongoing SAV proposals (e.g., BAR-SAV)
 - Quote from BAR-SAV draft: “The BAR-SAV algorithms in this document can easily accommodate the use of TOA if its adoption occurs in the IETF”

TOA eContent

```
RPKI-TOA-2026
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs9(9) smime(16) mod(8)
    id-mod-rpkiTOA-2026(TBD) }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS
  CONTENT-TYPE
  FROM CryptographicMessageSyntax-2010 -- in [RFC6268]
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(8) id-mod-cms-2009(58) } ;

ct-trafficOriginAttestation CONTENT-TYPE ::=
  { TYPE TrafficOriginAttestation
    IDENTIFIED BY id-ct-trafficOriginAuthz }

id-ct-trafficOriginAuthz OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) id-smime(16) id-ct(1) trafficOriginAuthz(TBD) }

TrafficOriginAttestation ::= SEQUENCE {
  version [0] INTEGER DEFAULT 0,
  asSet ASSET,
  ipAddrBlocks SEQUENCE (SIZE(1..2)) OF TOAIPAddressFamily }

ASSET ::= SEQUENCE (SIZE(1..10000)) OF ASID
ASID ::= INTEGER (0..4294967295)

TOAIPAddressFamily ::= SEQUENCE {
  addressFamily ADDRESS-FAMILY.&afi ({AddressFamilySet}),
  addresses ADDRESS-FAMILY.&Addresses
    ({AddressFamilySet}@addressFamily) }

ADDRESS-FAMILY ::= CLASS {
  &afi OCTET STRING (SIZE(2)) UNIQUE,
  &Addresses
  } WITH SYNTAX { AFI &afi ADDRESSES &Addresses }

AddressFamilySet ADDRESS-FAMILY ::=
  { addressFamilyIPv4 | addressFamilyIPv6 }

addressFamilyIPv4 ADDRESS-FAMILY ::=
  { AFI afi-IPv4 ADDRESSES TOAAddressesIPv4 }
addressFamilyIPv6 ADDRESS-FAMILY ::=
  { AFI afi-IPv6 ADDRESSES TOAAddressesIPv6 }

afi-IPv4 OCTET STRING ::= '0001'H
afi-IPv6 OCTET STRING ::= '0002'H

TOAAddressesIPv4 ::= SEQUENCE (SIZE(1..MAX)) OF TOAIPAddress{ub-IPv4}
TOAAddressesIPv6 ::= SEQUENCE (SIZE(1..MAX)) OF TOAIPAddress{ub-IPv6}

ub-IPv4 INTEGER ::= 32
ub-IPv6 INTEGER ::= 128

TOAIPAddress {INTEGER: ub} ::= BIT STRING (SIZE(0..ub))

END
```

A TOA can contain:

- A set of ASNs, and
- A set of IP prefixes

Every AS in the ASN set is authorized to source traffic from every prefix in the prefix set

Proposed Operational Considerations

□ Handling overlapping prefixes in TOAs (**discussed on mailing list after IETF124**)

◆ When multiple ASes are authorized for overlapping prefixes:

➤ Example:

➤ TOA1: {AS1, 203.0.113.0/24}

➤ TOA2: {AS2, 203.0.113.0/28}

◆ Both AS1 and AS2 are considered authorized for the more specific prefix (203.0.113.0/28)

◆ If the goal is to restrict AS1 from the more specific prefix, TOA1 must be modified to explicitly exclude that more specific prefix

Next Step

□ Discussion with Sriram

◆ Sriram: The draft mentions nothing about the need for an AS 0 ROA (if the prefix is not intended to be announced in BGP)

➤ We think this document **should only include TOA-specific operations and considerations**

➤ ROA-related operations and considerations are out of the scope of this document

□ Coordination with SIDROPS WG and seek feedback on eContent format

◆ Does the document need SAVNET WG adoption before going to SIDROPS WG?

Thanks