Questions

1)Marvin: How does security work for proving a HID is part of the AD? The Host must prove that its HID is valid by proving it is who it claims to be. Do the routers keep the necessary state necessary for verifying HIDs, or do the controllers keep it?

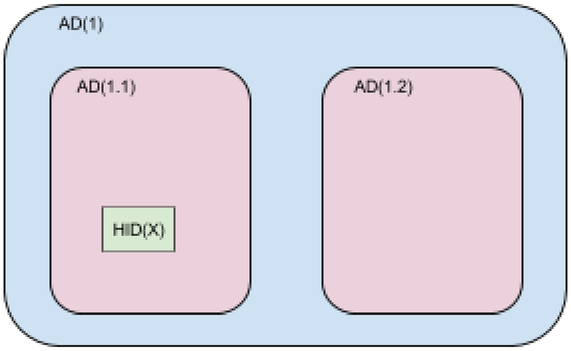
We leverage the concept from Root-of-Trust(RoT) in Scion to bootstrap security.
An AD managers the RoT file inside its own domain to manage it domain trust. RoT includes the controllers' certificates (public keys) and their self-signed signatures.
Bootstrap security inside an AD via registration:
1. a new device should install its own domain RoT manually (assumption).
2. the device registers its HID to one of the domain controllers; the RoT could guarantee the device could trust the controllers through challenge-response protocols
3. The controller signed the device's HID and notify all elements with update message (HID|| Sign(HID)_{Prikey})

Can we define 'TD Core' at the domain level instead of TD core? (Which is right?)

Accountable Internet Protocol section 5: http://mistlab.csail.mit.edu/papers/aip.pdf

2)Marvin: How can you use hierarchies to reduce the number of routing table entries? (our interpretation of the question)

Prefix-based aggregation is not possible in XIA with AD and HID principals, since there is no inherent hierarchy; AD and HID XIDs are essentially random, since they are hashes of public key. However, one can organise an AD hierarchically by divide it further based on sub-ADs. In this case, the destination DAG would, for example, be like [• → AD(1) → AD(1.1) → HID(x))]. AD(1) will only advertise itself to other ADs and not its sub-ADs, so routers in other ADs will only have entry for the top-level AD, (i.e., AD(1)) in their forwarding table.



3)Who constructs XION packets

- Marvin: Hosts can't possibly be responsible for this, must be gateway
- Which is simpler? Do that…

* What would clients send if done at the gateway?

Current solution: The gateway caches all up- and down-paths information for XION. After a host queries a destination HID and received the AD via DNS service, the host constructs it data packet by adding its source/destination DAG, forward it to the gateway, and the gateway will process the packet (e.g., add XION path to the extension header if the host "wants" to use XION). The question here is how the host express her interest to the gateway, such like " I want to use XION instead of XBGP". This part could be designed by inserting different principal types into the destination DAG.

Research question: What's the right amount of "selection power" to expose to an end client? There might be some policy research to be done here. Can clients ask for SCION vs. BGP?

Raja:
4)How does XIA make it easier to add new services or change within a domain w/o anything outside the domain knowing anything about the change? Can we abstract the details of the implementation of a particular service so that implementation changes don't affect how hosts in other domains communicate with the service?

* Examples: routing service within domain is changed from a single machine implementation to a fault-tolerant one w/Paxos. Can this be done w/o
external entities who use this service knowing anything about it?

What should be encoded in a DAG?
5)Modify the DAG vs. encapsulation:

* Marvin: Modifying the DAG is better because its what's difference about XIA
* ?Aditya?: Encapsulation preserves intent…
* Srini: Modifying DAG might have security implications

Should the destination DAG always remain the same at the destination as it was defined by the receiver?

- **What is the true semantic meaning defined by the "destination DAG"**

More general: What are the tradeoffs between encapsulation vs. modifying the DAG.

* Does Serval do some of this?

6)Extension headers vs. in DAG

* Srini: Can SCION Paths have fallbacks? Peter: No, doesn't make sense in the general since, though some XID types in fallbacks might work (e.g., non-scoped SIDs)

Again, what is the meaning of a DAG? Also variable-length XIDs?