# Multihop Federations
## draft-mrw-abfab-multihop-fed-01.txt

Margaret Wasserman
mrw@painless-security.com

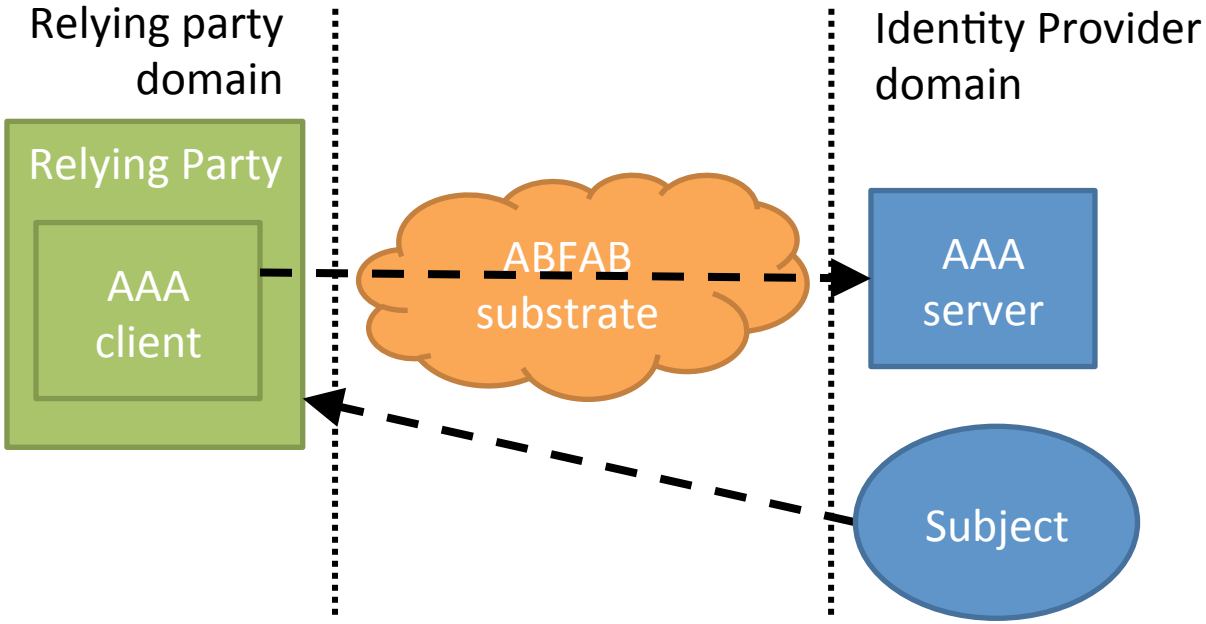
# Why Am I Here?

- Presenting work that has been proposed in the ABFAB WG for Multihop Federations
  - Overall Multihop Architecture and Trust Router
    - draft-mrw-abfab-multihop-fed-01.txt
  - Key Negotiation Protocol
    - draft-howlett-radsec-knp-01.txt
- Not changing AAA protocols
  - Specifications compatible with Radius, RadSec and Diameter
- Here to get your feedback/comments

# ABFAB Architecture

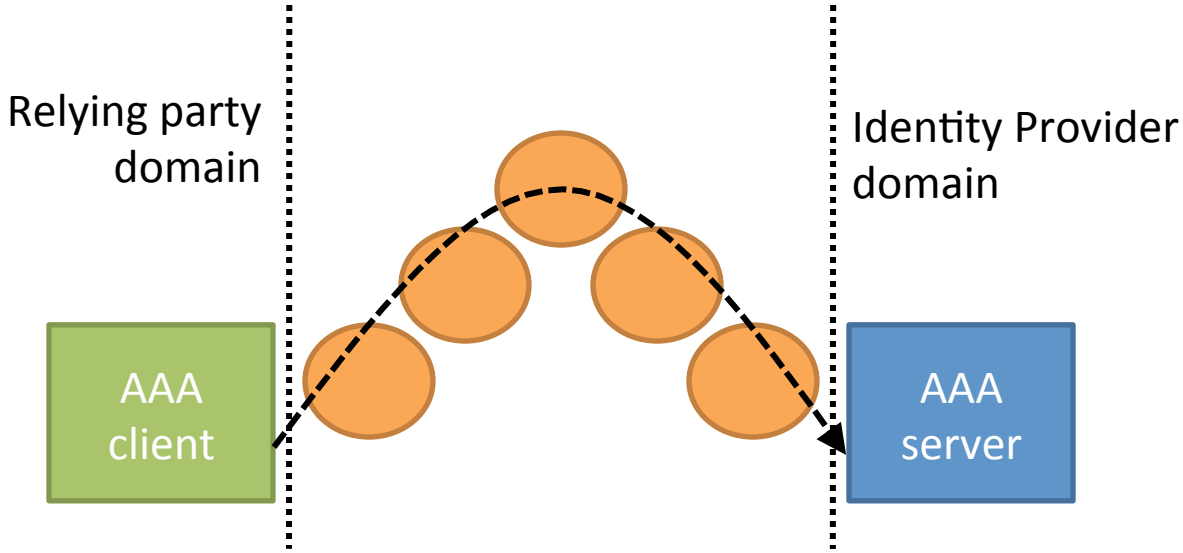
- **ABFAB – Application Bridging for Federated Authentication Beyond the web**
- **ABFAB architecture is described in**
  - draft-lear-abfab-arch-02.txt
- **ABFAB allows the use of AAA protocols for application authentication in non-Web apps**
  - **Makes use of GSS-EAP and EAP Channel Bindings**
- **Subject may use a single EAP credential, from his Identity Provider, to authenticate to multiple applications within the federation**

# ABFAB architecture



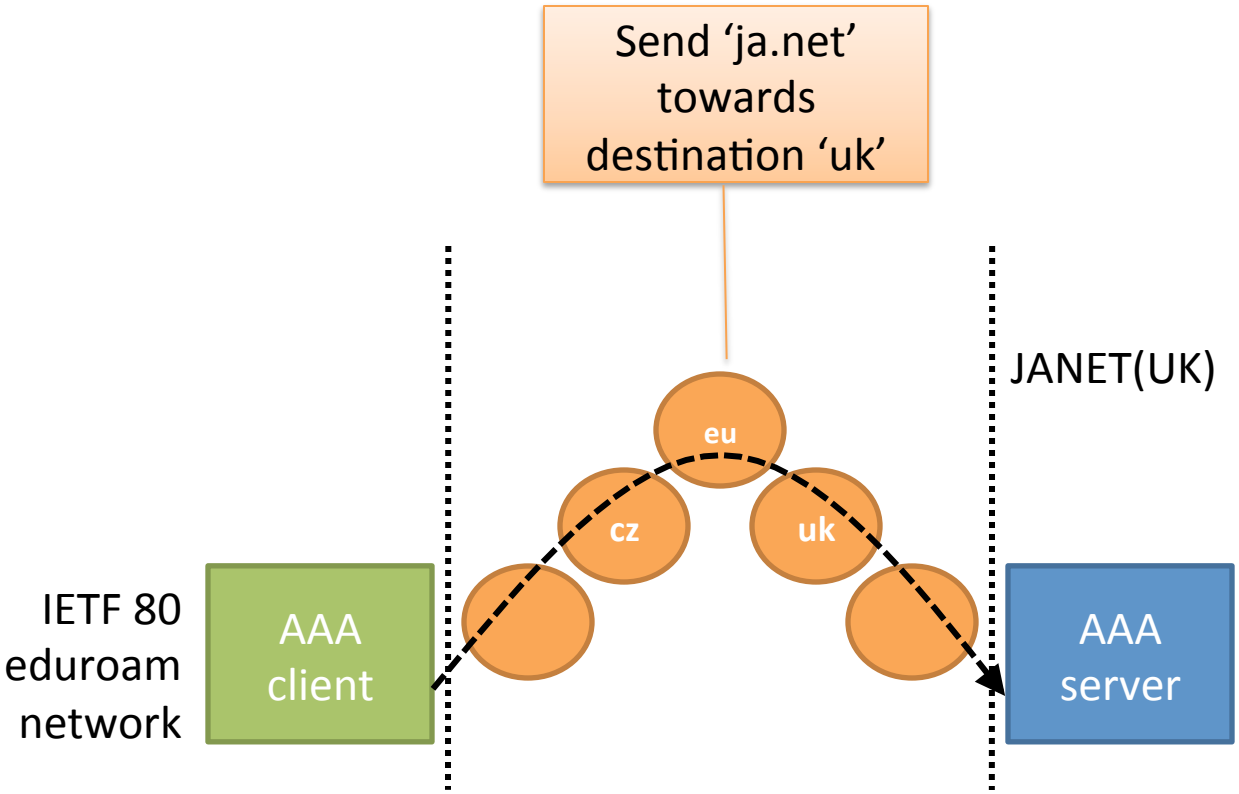

- The **ABFAB substrate** provides four functions:
  - **Transport**: how messages are conveyed between client and server
  - **Server discovery**: how Relying Parties find a server in the Subject's domain
  - **Trust establishment**: how the client/server establish confidence that they are talking to the right client/server.
  - **Rules determination**: how the client/server decide what they should infer from the messages, and how they should behave in that regime.

# RADIUS Proxy Substrate

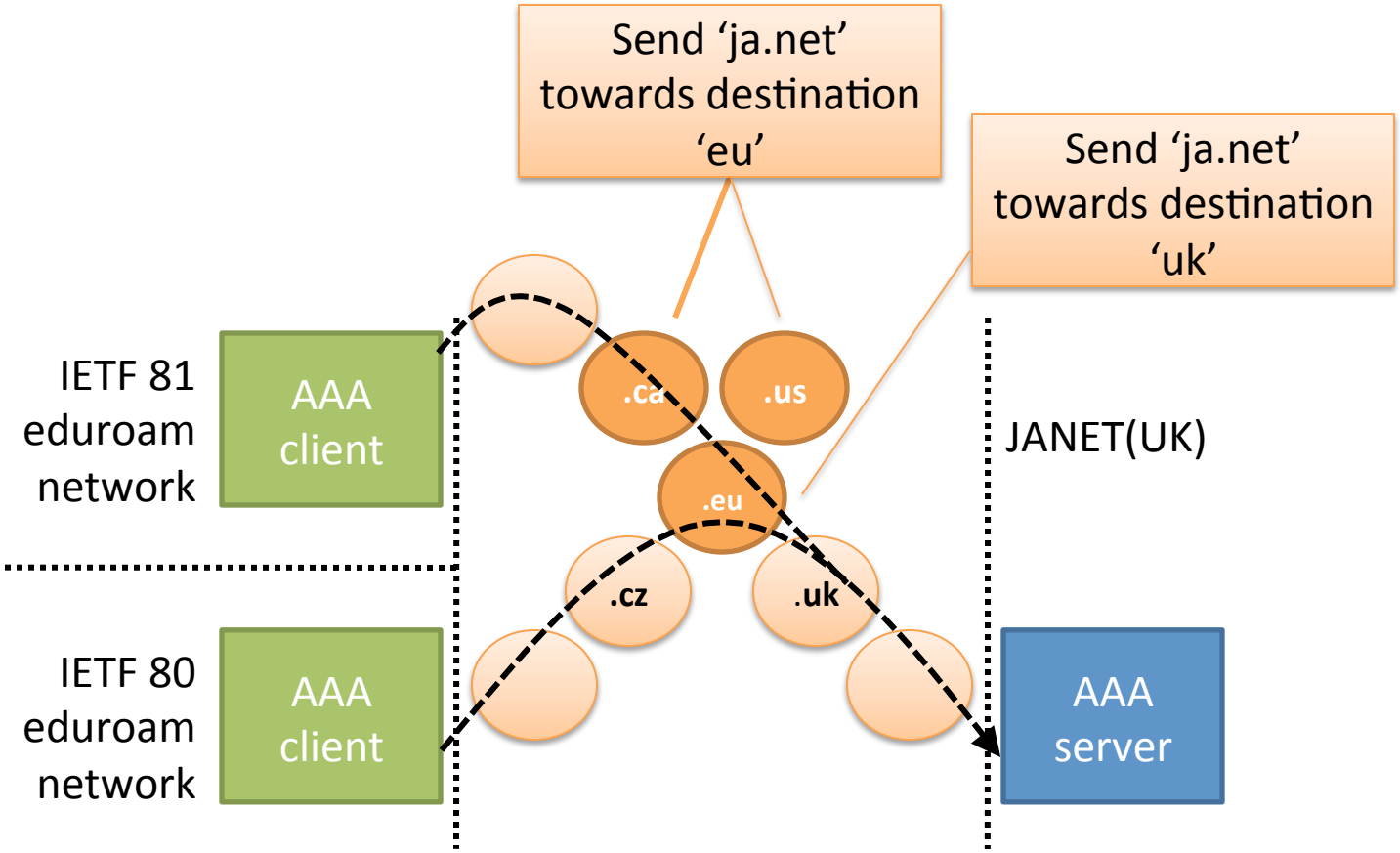



- An authentication request will traverse a set of AAA proxies
- Each AAA proxy knows how to forward the request
  - Based on destination realm or hierarchical portion of the realm

# Static configuration is simple…



# …until it isn't.



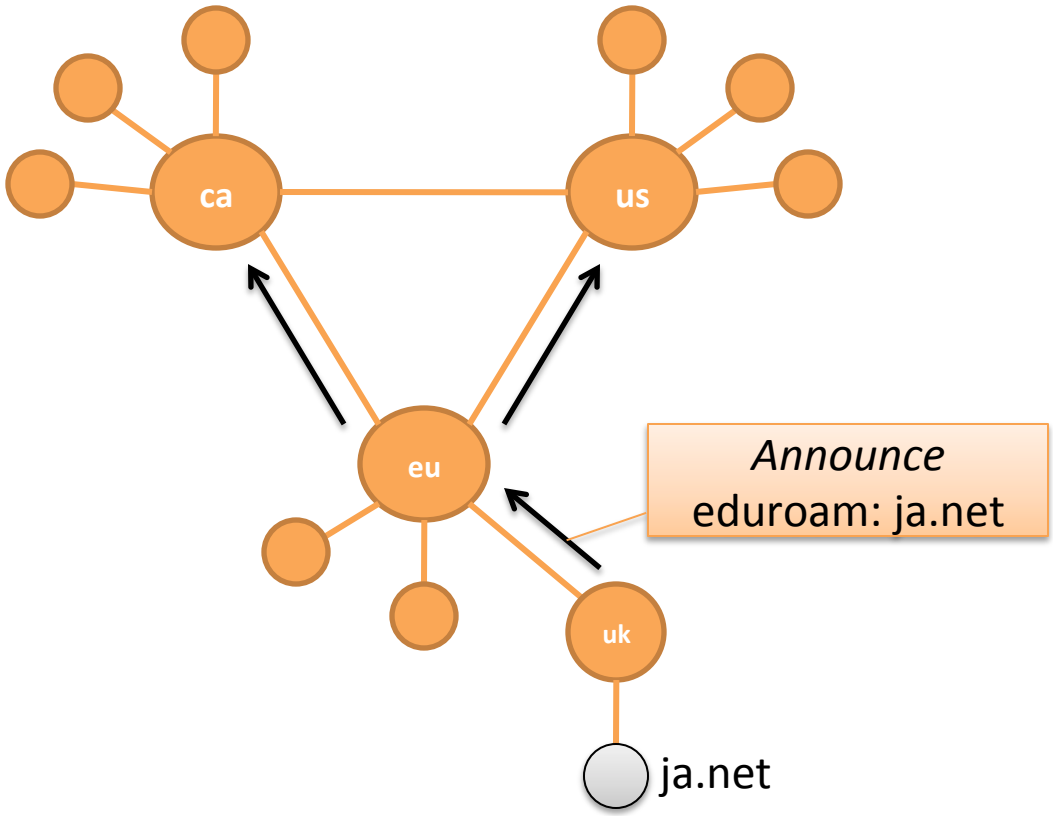
# Static configuration doesn't scale

- As an AAA system scales, you need to maintain more configuration across more nodes.
- The configuration is necessarily dissimilar between AAA nodes, but the entire system needs to behave as though all nodes share a consistent view of the entire system. Inconsistency may result in undesirable behaviour.
- Inventing an *ad hoc* solution within a single domain is trivial. The multi-domain case is also tractable, providing there is close coordination.
- However, if ABFAB is successful the potential number of domains and overall system size is considerable: coordination will be challenging.
- We need a standard mechanism that enables AAA nodes within a large and loosely-coupled AAA system to behave as though they share a consistent view of the entire system.

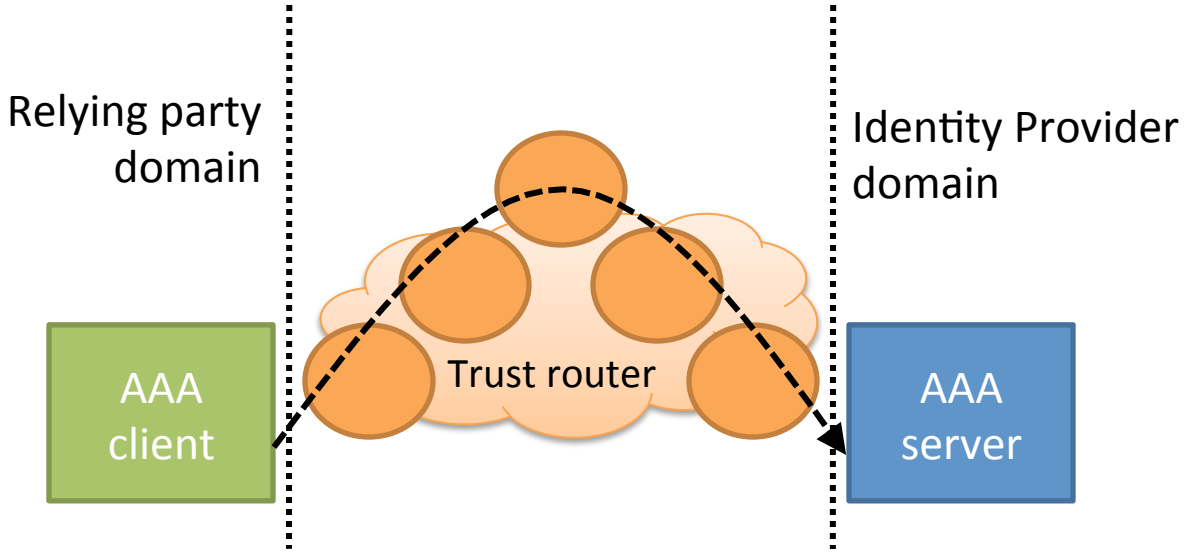
# Introducing the Trust Router

- Serves a similar role to BGP in IP routing
  - Distributes information about available “Trust Links” within a federation (or “Policy Regime”)
  - Calculates a local tree of “Trust Paths” to reach destination realms
  - Determines the “best” path to reach each destination realm

# Trust router protocol



# RADIUS substrate




- **Trust Router allows path selection through the AAA fabric**
- **But, static configuration is still required at each hop for trust establishment**
- **All AAA servers in the path can see session keys and, potentially, personal information such as real names**

# Well, we have RadSec…

- RadSec is Radius over TLS or DTLS
- Invoke PKI to banish hop-by-hop security; permits e2e trust establishment
- Knowing your peer explicitly may improve rules determination

- Other benefits:
  - Prevents exposure of information to intermediate AAA nodes
  - Reduces EAP transmission latency

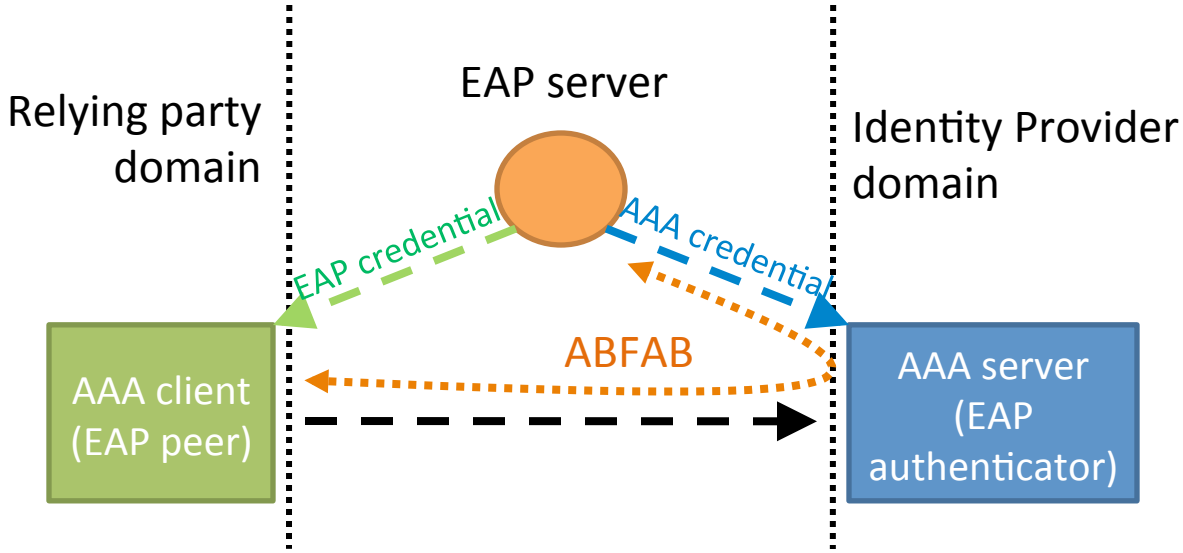
# A single PKI for ABFAB deployments?

- A PKI environment is a one-to-many relationship; good when you have uniform business requirements and a small number of certificate authorities
- However, a one-to-many relationship imposes costs (financial and operational) on all Relying Parties that may not match varied business requirements
- A one-to-one relationship allows the actors to agree to their mutual business requirements
- But pairwise credentials don't scale, right?

# Didn’t we just fix the multiple credential problem?

- We’ve just described a mechanism (ABFAB) that enables a single EAP credential to be used with all RPs that trust the EAP server
- An AAA server is just another RP, so let’s apply ABFAB to RadSec!

# RadSec with ABFAB




- **Allows trust to be established using ABFAB, not PKI**
- **However, not all AAA clients and AAA servers in a large federation will be connected via a single EAP server**

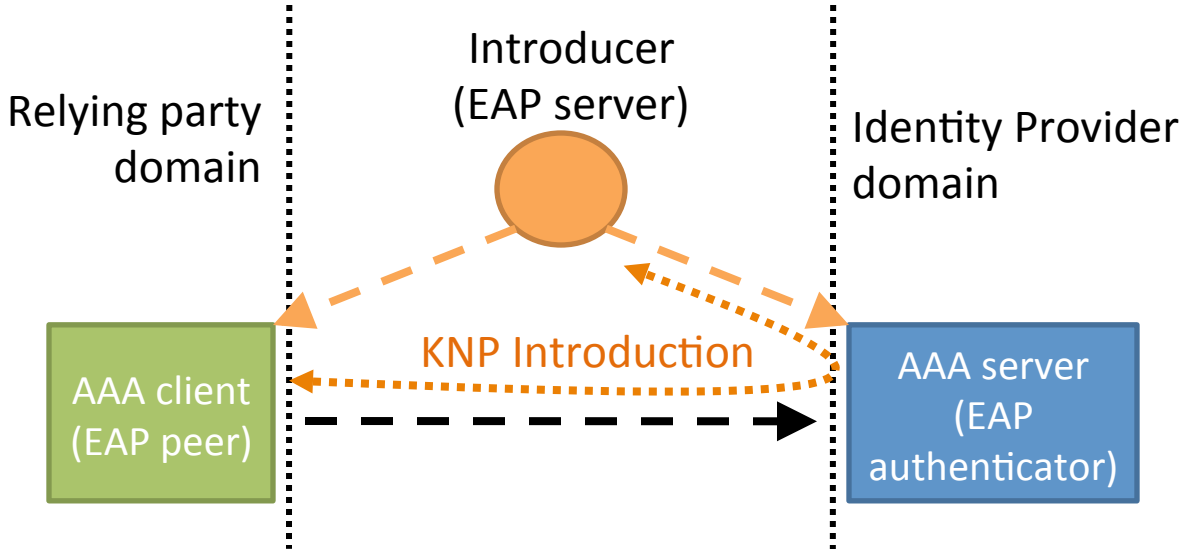
# Key Negotiation Protocol

- KNP enables a RadSec client and server to dynamically establish a short-lived credential for a subsequent RadSec connection.

- KNP uses EAP authentication of credentials issued to the AAA client by an EAP server that is also trusted by the AAA server.

- The EAP server is called the ‘Introducer’. The process of establishing the RadSec credential between AAA client and server is called ‘Introduction’.

# KNP Introduction

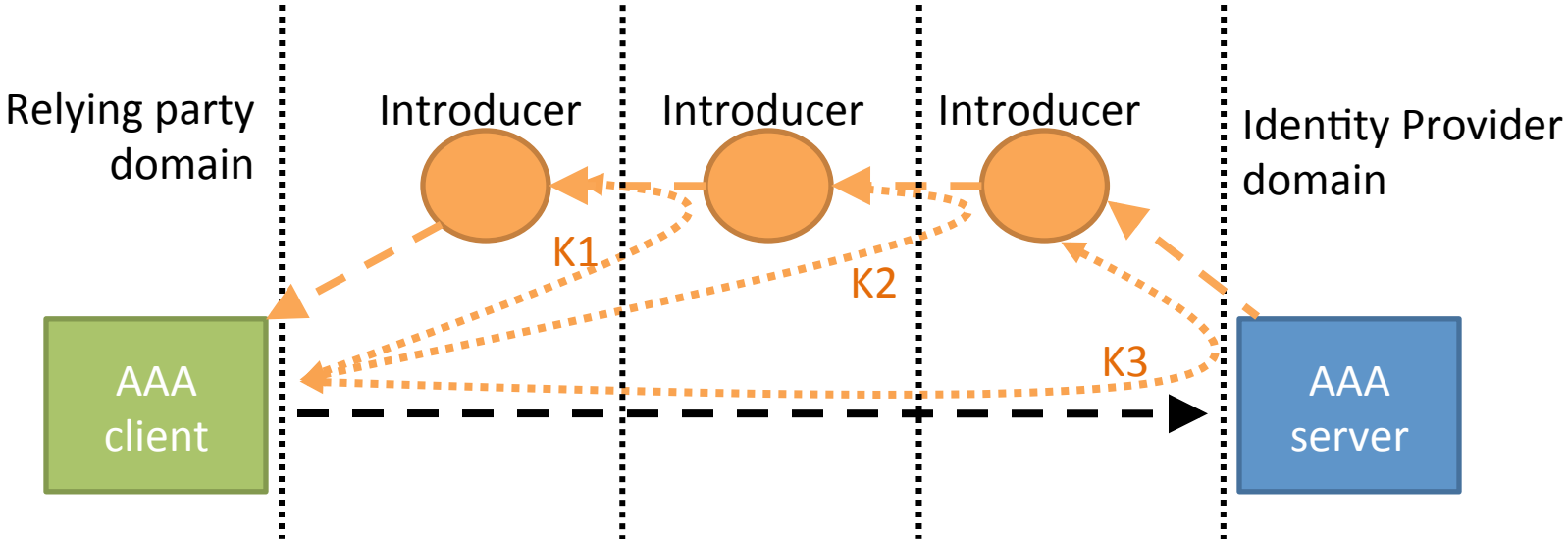



**When an AAA Client and a AAA Server are connected via a single KNP Introducer, this is referred to as a Trust Link**

# Transitive operation

- Not all AAA nodes share a common Introducer.
- An Introducer can also be party as AAA client or server to an Introduction.
- This enables transitive introduction: the AAA client recurses along a path of Introducers to the AAA server.

# Transitive Use of KNP




- **When a AAA Client can reach a AAA Server through a chain of KNP Introducers, this is a Trust Path**
- **How does the RP know what path to traverse? It asks it's local Trust Router!**

# Trust Path

- A Trust Path is a series of KNP hops that can be used to reach a AAA server in a destination realm
- Each KNP hop is called a Trust Link
- Shown as series of realms and types, connected by arrows
  - Currently defined types are Trust Router (T) or AAA Server (R)
  - Example: A -> B(T) -> C(T) -> D(T) -> D(R)

# Trust Router Functions

- Trust Router Protocol
  - Distributes information about available Trust Links in the network
  - Calculates a tree of Trust Paths to reach target destinations
- Trust Path Query
  - Provide “best” path to a destination realm in response to queries from local RPs
- Temporary Identity Request
  - Provision temporary identities that RPs can use to reach the next hop in the Trust Path, in response to KNP requests from RPs
  - AKA, serve as a KNP Introducer

# Trust Router Protocol

- Exchange information about Trust Links between Trust Routers
  - Trust Links are unidirectional and of a specific type
    - A -> B(T) does not imply A -> B(R), B -> A(T) or B -> A(R)
  - Realm names are not necessarily hierarchical, but they may be
    - example-u.ac.uk is not necessarily reached via .uk or .ac.uk
- Tree of available Trust Paths rooted in local realm is calculated by each Trust Router

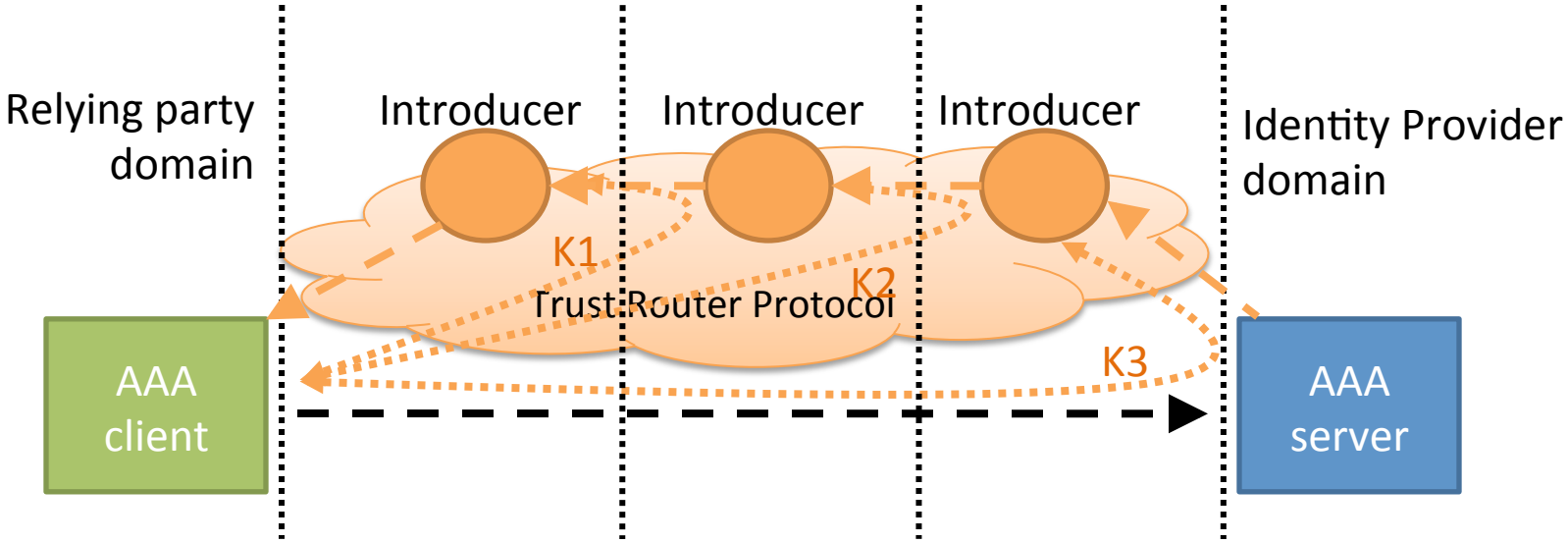
# Trust Path Query

- Generated by an RP to request a Trust Path to reach a AAA server in a destination realm
- When a Trust Path Query is received, the Trust Router:
  - Authenticates the RP, and checks local policy to determine whether or not to reply
  - Searches its tree of Trust Paths to find the best path to reach the destination
  - Returns the best path, if found, to the RP

# Temporary Identity Request

- The RP issues a Temporary Identity Request to obtain an identity that will be used to traverse each link in the Trust Path
  - The existence of the Trust Link implies that a Temporary Identity Request will be granted

# ABFAB Multihop Federation




- **Uses ABFAB, KNP and Trust Routers to allow RPs to reach AAA Servers in all destination realms that can be reached through a transitive Trust Path across the federation**
- **Minimal per-hop configuration, as needed to define one-to-one trust relationships and express local policy**

# Questions? Feedback?

- Questions about what we are proposing?
- Feedback on this proposal?

- discussion to abfab@ietf.org

# BACKGROUND SLIDES

# Concerns about PKI for ABFAB

- PKI makes sense where you have uniform business requirements and a small number of certificate authorities (ideally one)
- However, ABFAB federations are often composed of entities with different security requirements
- Multiple trust authorities may be needed to support certification within regional, legal or organizational boundaries.
  - To comply with different local laws
  - To allow local authorities within a country or continent
  - Some organizations may demand local control

# Multiple Business Requirements

- May require multiple types of certificates
  - Financial costs (to purchase certificates, software, etc.)
  - More complex, longer registration/enrollment, limited by CA policies
  - Increased administrative and support complexity (e.g. knowing which certificates are valid for what)
- Or force fit all requirements to a single certificate type
  - Match lowest security requirements, to reduce costs
    - May compromise security for RPs with higher security requirements
    - Lower than ideal security
      - - e.g. People may use existing certificates for new applications, even when they aren't a good fit for the security requirements
  - Or match highest security requirements
    - Imposes higher then justified cost on RPs with less stringent security requirements
    - More complex, longer registration/enrollment, limited by CA policies
    - Some RPs may not be able to meet stringent requirements, which leads to lower than ideal security
      - e.g. People may bypass the PKI for things

# Multiple CAs

- High cost to establish procedures for cross/multi-CA trust
  - Establishing cross-CA policy is time-consuming and expensive
  - May include requirements for cross-CA auditing
- Leads to more complex, more costly registration procedures
  - May be union of security requirements of all CA
  - Some RPs may not be able to meet stringent requirements, which leads to lower than ideal security
    - e.g. People may bypass the PKI for things that don't fit the CA policies
- Security of the overall system depends on the weakest link

# Why is Trust Router/KNP Better?

- Each Trust Router peering is a separate business relationship
  - Relationship is negotiated between two parties
  - Parties can control their own costs