Secure Service Discovery in open Networks

Author:
Markus Alexander Kuppe
Vitali Amann

http://github.com/lemmy/SecuredSLP

Outline

- Motivation
- Open network
- Discovery architectures
- Service Location Protocol (SLPv2)
  - Thread analysis SLPv2
- Secure SLP
  - Trust scenarios in open networks
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

Use Case

Outline

- **Motivation**
- Open network
- Discovery architectures
- Service Location Protocol (SLPv2)
  - Thread analysis SLPv2
- Secure SLP
  - Trust in open networks
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

Motivation

- Create a secure network
- Mechanism for
  - providing services
  - sharing services
  - discovering services
- Prevent or complicate exploitation
  - tracking user agents
  - manipulate service information
  - replaying attacks

Outline

- ☑ Motivation
- **Open network**
- Discovery architectures
- Service Location Protocol (SLPv2)
  - Thread analysis SLPv2
- Secure SLP
  - Trust in open networks
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

Open Network

- A network for everyone (even for bad guys)
- Networking everywhere
- Not necessarily based on IP (but here)
- Contains several devices
  - Notebooks
  - Mobile phones
  - PDAs
  - Printer
  - and other portable or stationary devices
- Share services

[P2PFound]

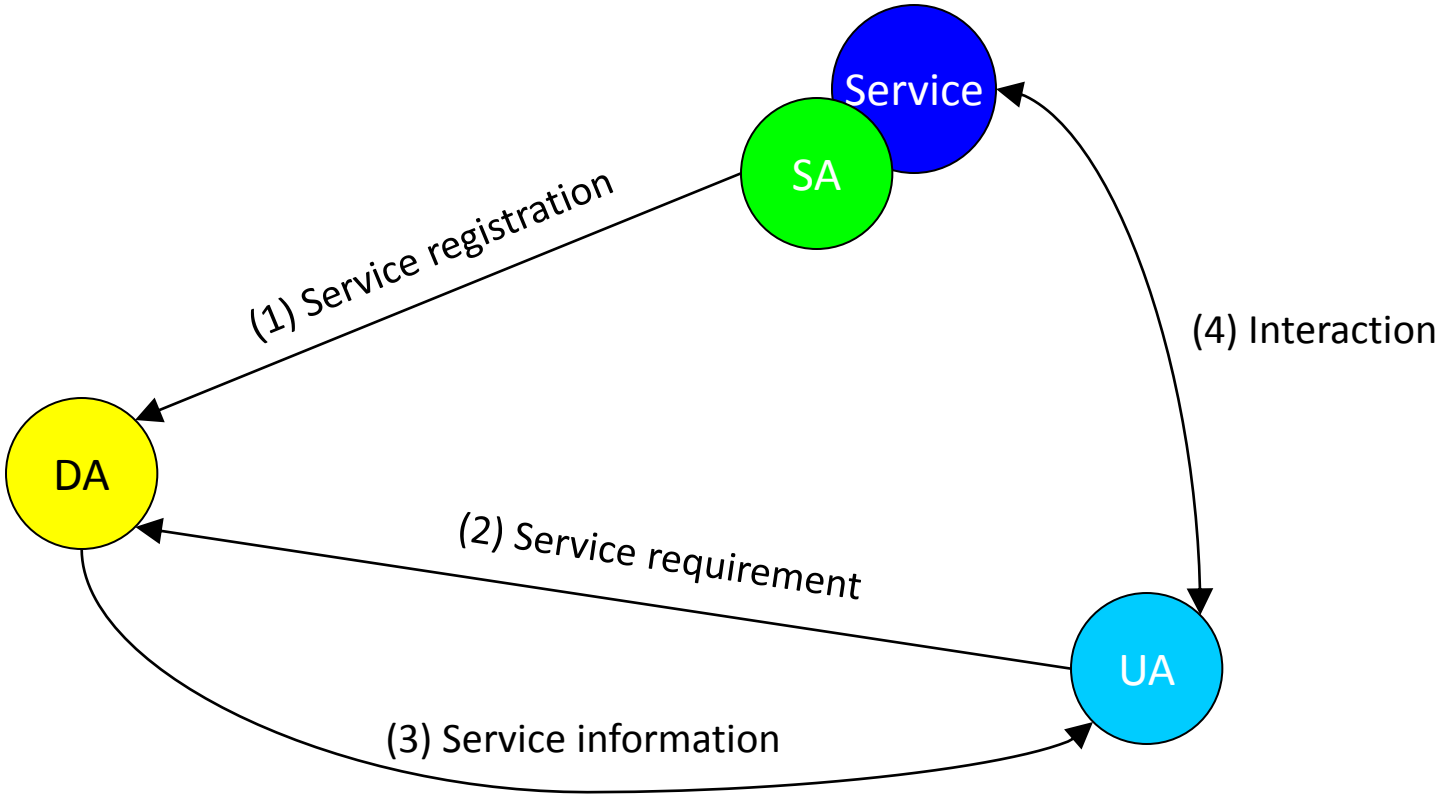
Outline

- ☑ Motivation
- ☑ Open network
- **Discovery architectures**
- Service Location Protocol (SLPv2)
  - Thread analysis SLPv2
- Secure SLP
  - Trust in open networks
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

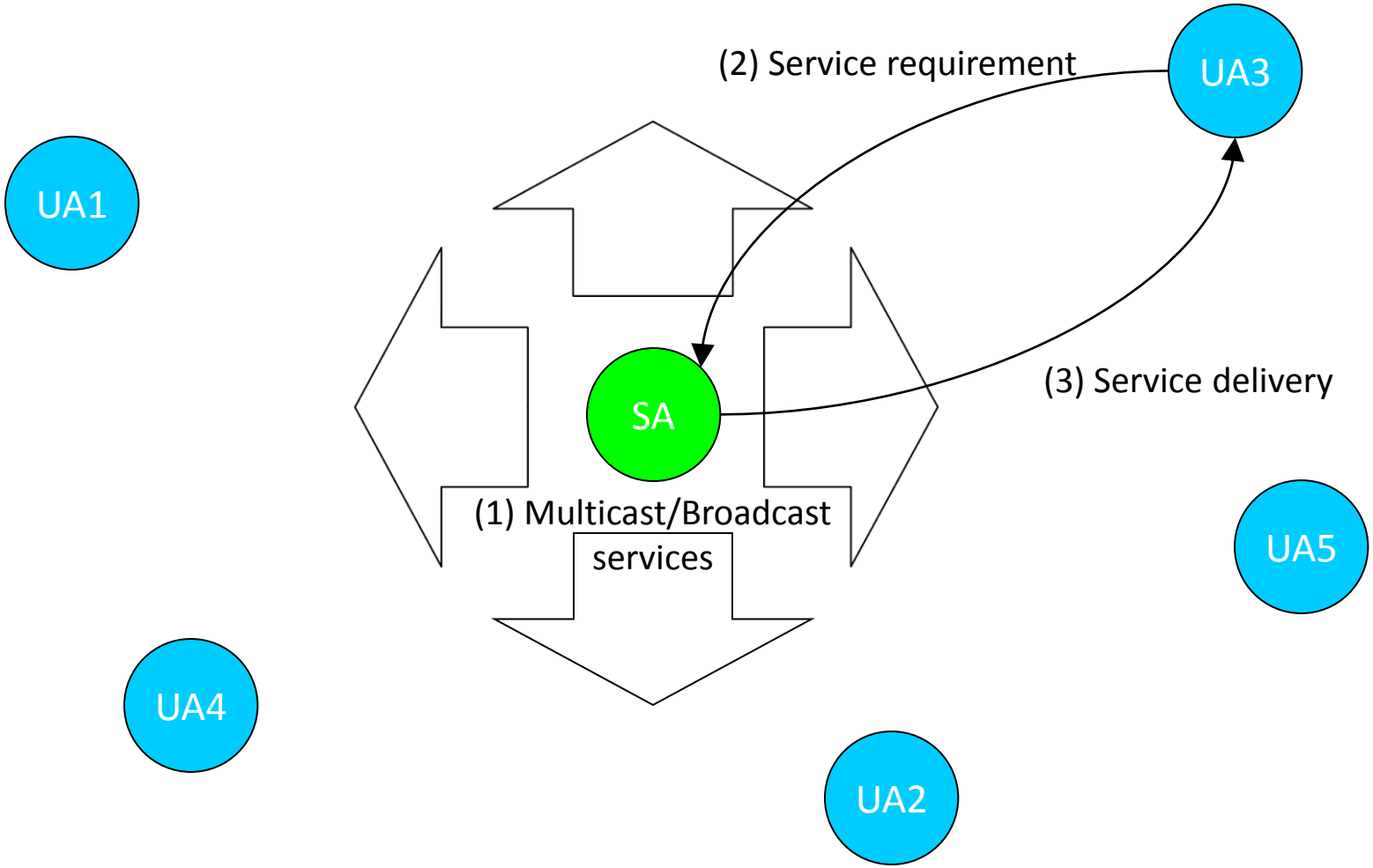
Discovery architectures

- Three main architectures
  - Directory-based architecture
  - Directory-less architecture
  - Hybrid architecture
- Three possible actors
  - Service agent (SA)
  - User agent (UA)
  - Directory agent (DA)

Hybrid architecture: Case 1



Hybrid architecture: Case 2



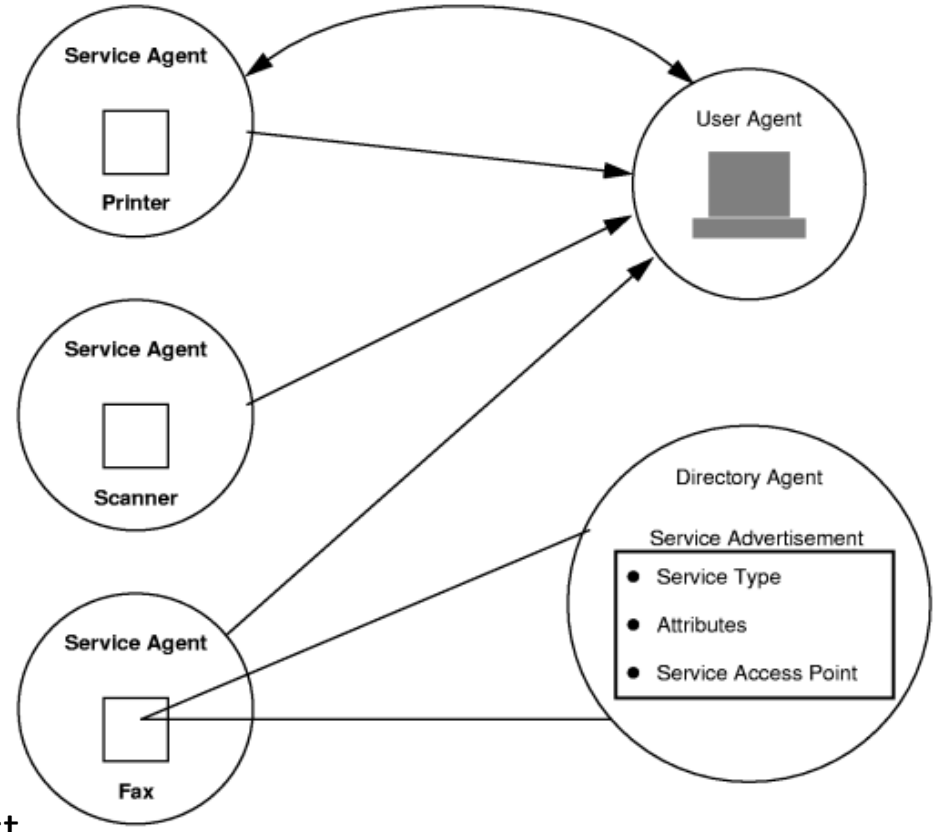
Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- **Service Location Protocol (SLPv2)**
  - **Thread analysis SLPv2**
- Secure SLP
  - Trust in open networks
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

Service Location Protocol (SLPv2), RFC 2608

- **Multicast** discovery
- **Unicast** answers (UDP & TCP)

- Seamless transformation from
  - Multicast convergence to
  - Directory Agent (DA)
    - DA discovery still multicast
- Partitioning via application layer
  - **Scopes**
- Trust with **pre-established asymmetric keys**
  - no support for “dynamic” trust



Thread analysis for traditional SLPv2

Confidentiality	no
Integrity	yes
Authentication	yes
Authorization	no
Replay prevention	no
Availability	no
Non-repudiation	no

Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- ☑ Service Location Protocol (SLPv2)
  - ☑ Thread analysis SLPv2
- **Secure SLP**
  - **Trust in open networks**
  - Security Groups
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

Trust in open networks

- Web of trust
- Public Key Infrastructure (PKI)
- (Reputation based identity)

Web of trust

- If you trust A then you trust everyone trusted by A
- Decentralized structure
  - no extra server needed
- Based on public-key cryptography
- Sign keys which you trust

Public key infrastructure (PKI)

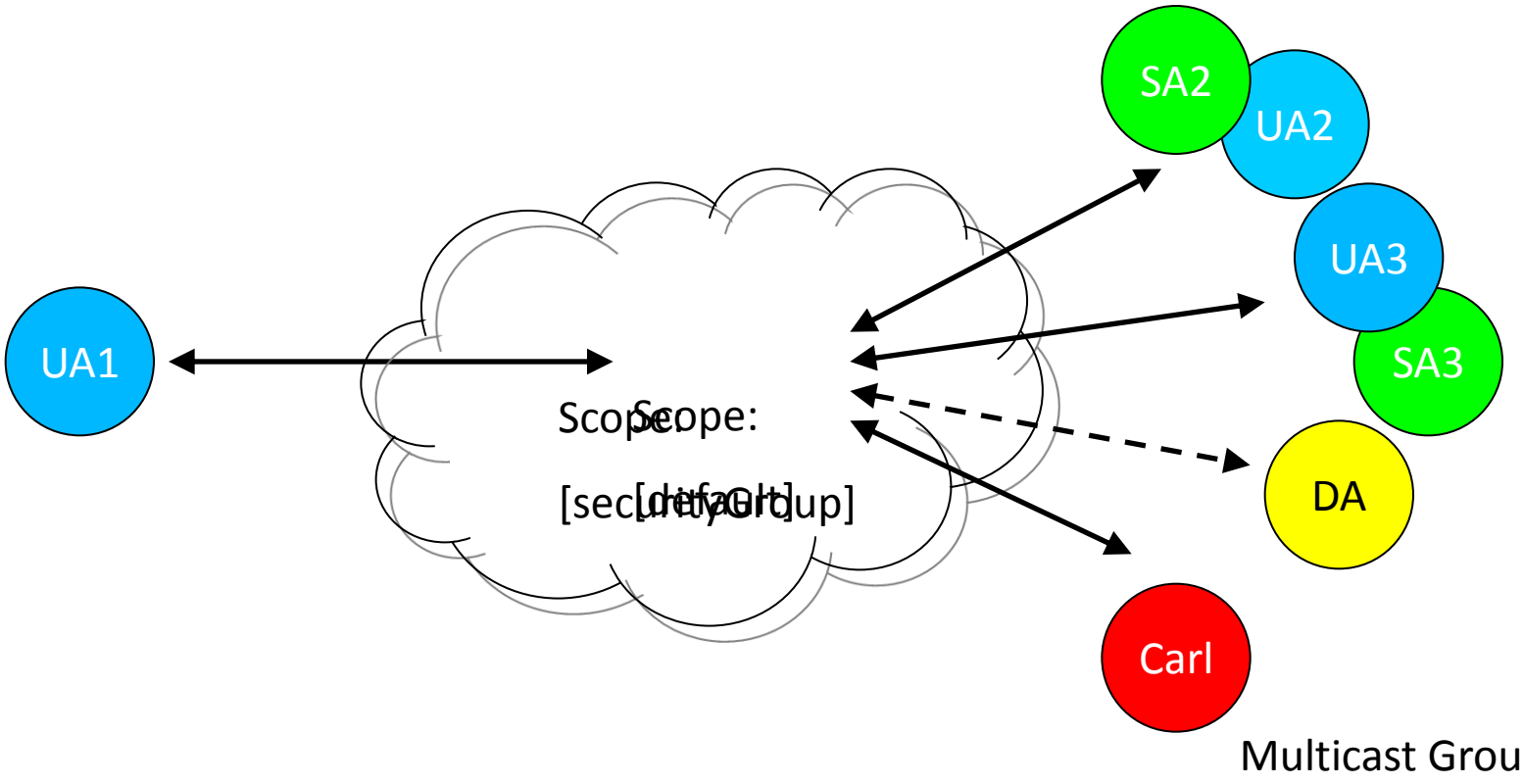
- Requires several instances
  - Registration authority
  - Certificate authority
  - Validation authority
- Based on public-key cryptography
- Better with internet access
- Centralized structure
  - Requires a server

Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- ☑ Service Location Protocol (SLPv2)
  - ☑ Thread analysis SLPv2
- Secure SLP
  - ☑ Trust in open networks
  - **Security Groups**
    - Group Diffie-Hellman
  - Thread analysis SecureSLP
- Conclusion & Future Work

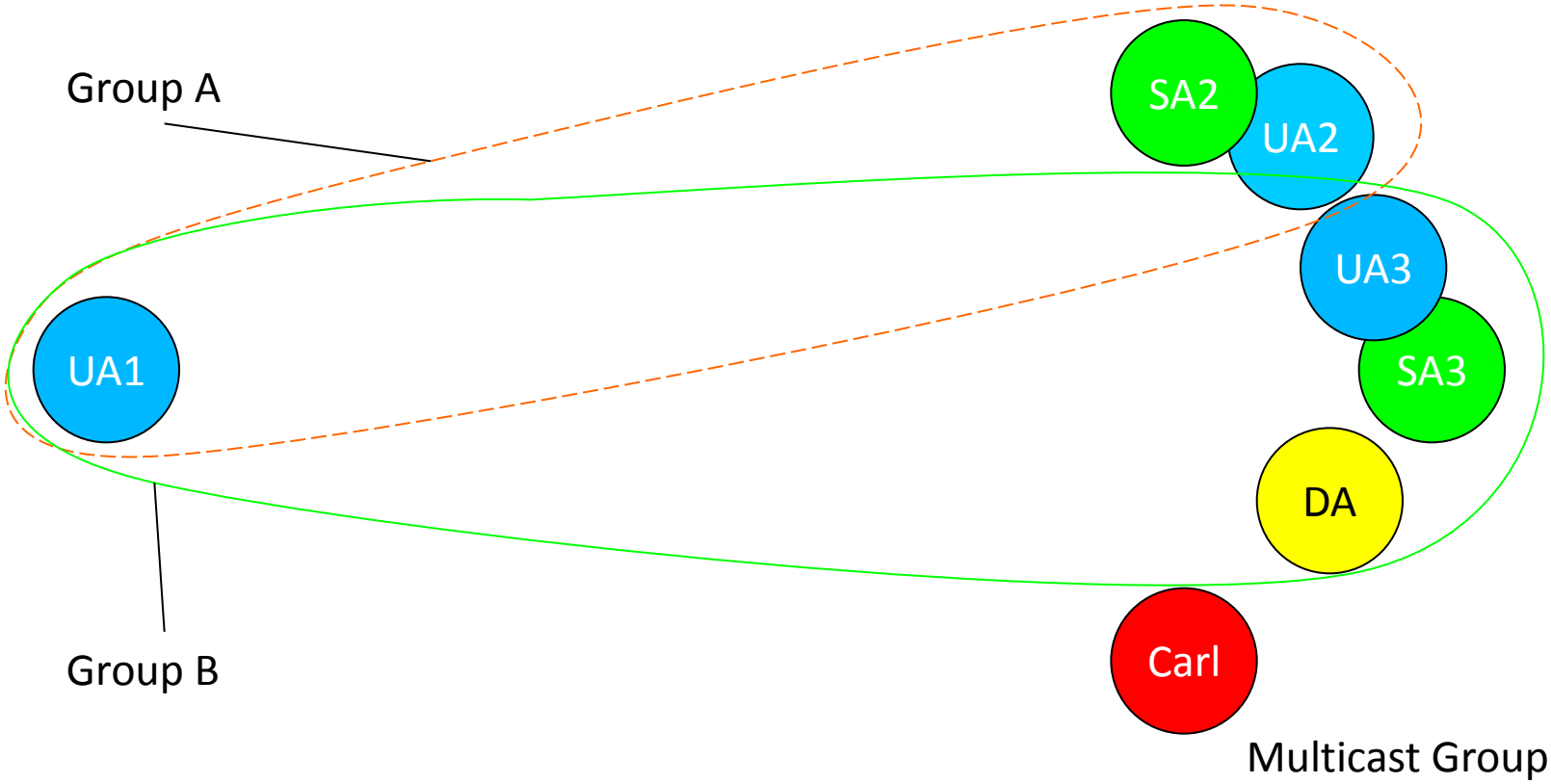
Security Groups (SG)

- SGs create a confidential channel among trusted peers (trust is pre-requisite)



Security Groups (SG)

- SGs create a confidential channel among trusted peers (trust is pre-requisite)



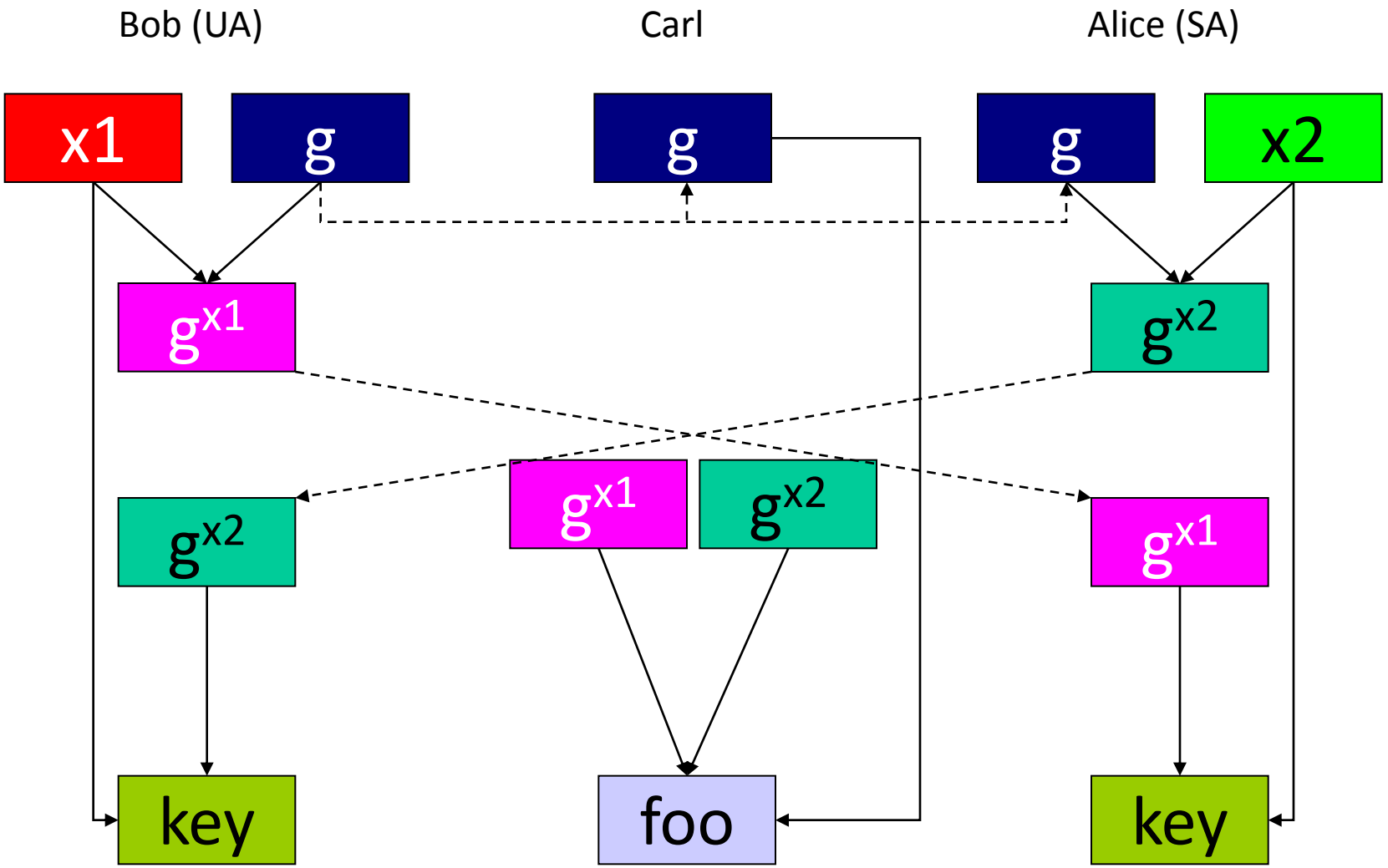
Security Groups (SG)

- Requirements
  - unsecure channel
  - decentralized architecture (?)
  - peers do not know each other (?)
  - dynamic group membership
  - all participants agree on the same key
  - key independence
    - no past/future data is allowed to be decrypt-able by future/past group members
    - Can be relaxed due to without SA joining, SG‘s data is stale
- ➢ SGs boil down to Group Key Agreement protocol (GKA) like Group Diffie-Hellman

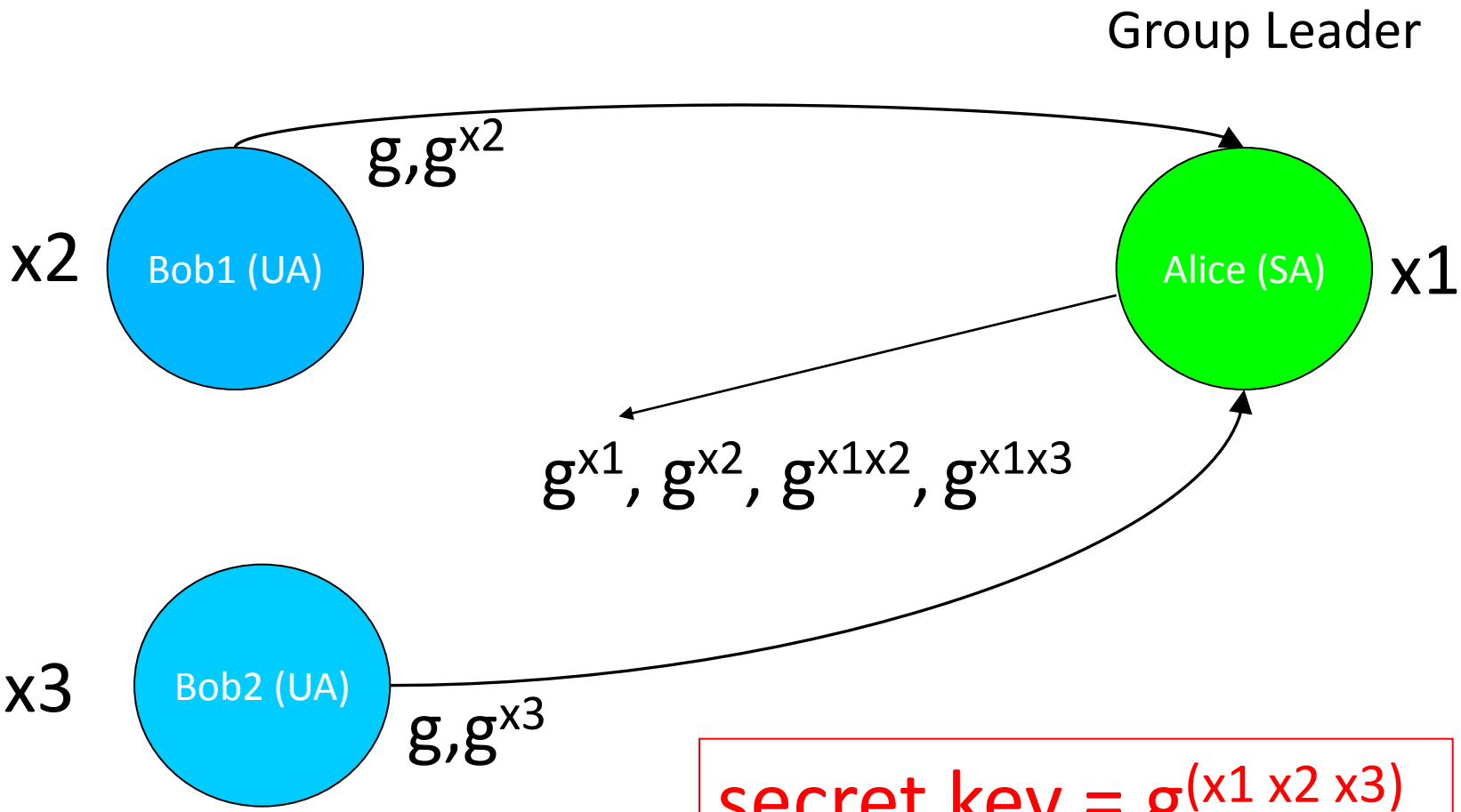
Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- ☑ Service Location Protocol (SLPv2)
  - ☑ Thread analysis SLPv2
- Secure SLP
  - ☑ Trust in open networks
    - Security Groups
      - **Group Diffie-Hellman**
    - Thread analysis SecureSLP
- Conclusion & Future Work

Diffie-Hellman RFC2631 recap



Asymmetric Group Diffie-Hellman (AGDH)



Asymmetric Group Diffie-Hellman (AGDH) cont.

- Key is composed of each peers contribution
  - contrary to GKE where one peers sends the key to all participants
- Join/Leave demands re-keying
  - peers contribution is added/removed from the group key
- One affects all
  - compromise of a single group member affects the security of the whole group
- (Perfect) Forward Secrecy (PFS)
  - Compromise of long-term keys cannot result in the compromise of past session keys

Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- ☑ Service Location Protocol (SLPv2)
  - ☑ Thread analysis SLPv2
- Secure SLP
  - ☑ Trust in open networks
  - ☑ Security Groups
    - ☑ Group Diffie-Hellman
  - **Thread analysis SecureSLP**
- Conclusion & Future Work

Thread analysis SecureSLP

	SLP	SecureSLP
Confidentiality	no	yes
Integrity	yes	yes
Authentication	yes	yes
Authorization	no	?
Replay prevention	no	yes
Availability	no	no
Non-repudiation	no	no

Outline

- ☑ Motivation
- ☑ Open network
- ☑ Discovery architectures
- ☑ Service Location Protocol (SLPv2)
  - ☑ Thread analysis SLPv2
- ☑ Secure SLP
  - ☑ Trust in open networks
  - ☑ Security Groups
    - ☑ Group Diffie-Hellman
  - ☑ Thread analysis SecureSLP
- **Conclusion & Future Work**

Conclusion & Future Work

- (SLP's architecture qualifies it for use in open networks)
  - hybrid architecture
- SLP's security features are not up to speed with open networks
  - trust model is static
  - no confidentiality
- SecureSLP adds missing security features while staying protocol compatible to SLPv2
- Group Diffie-Hellman a good candidate for security groups
  - all peers knowing of each other might be too expensive

- What about *authorization, non-repudiation, availability?*
- Proof-of-concept implementation
  - Asymmetric Group Diffie-Hellman & additions to SLP

Questions

References

[NetIP, Inc.] - http://www.netip.com/articles/keith/diffie-helman.htm

[P2PFound] - http://p2pfoundation.net/Free_and_Open_Network_Definition