PriSrv: Privacy-Enhanced and Highly Usable Service Discovery in Wireless Communications (NDSS 2024)

Yang Yang, Robert H. Deng, Guomin Yang, Yingjiu Li, HweeHwa Pang,

Minming Huang, Rui Shi, Jian Weng

1. School of Computing and Information Systems, Singapore Management University, Singapore
2. Department of Computer Science, University of Oregon, USA
3. Beijing Electronic Science and Technology Institute, Beijing, China
4. College of Information Science and Technology, Jinan University, Guangzhou, China

Service Discovery

- **Service discovery protocols (SDPs)** are essential components of networking systems
  - They enable devices and services to **dynamically discover** and communicate with each other in a network environment
  - They facilitate the **automatic detection and advertisement** of available services, making it easier for devices to locate and interact with desired resources
  - Well known SDPs includes

    Wi-Fi, AirDrop, BLE, DNS-SD, mDNS, SSDP, UPnP, etc.

Attacks on SDPs

SDPs	Man-in-the Middle (MitM) Attacks	Spoofing Attacks	Denial-of-service (DoS) Attacks	User Identification Attacks	Tracking Attacks
DNS-SD [18]	√				
mDNS [19]	√				
SSDP [20]	√				
UPnP [21]			√		
Wi-Fi [1]	√	√		√	√
BLE [3]	√	√		√	√
AirDrop [2]	√	√		√	
PrivateDrop [16]				√	
CBN [9]	√	√			
WTSB [5]				√	√

[16] A. Heinrich, M. Hollick, T. Schneider, M. Stute, C. Weinert. PrivateDrop: practical privacy-preserving authentication for Apple airDrop. In USENIX Security, 2021.

[9] A. Cassola, E. O. Blass, G. Noubir. Authenticating privately over public Wi-Fi hotspots. In CCS, 2015.

[5] D. J. Wu, A. Taly, A. Shankar, D. Boneh. Privacy, discovery, and authentication for the internet of things. In ESORICS, 2016.

SDPs: Requirements

Privacy Enhancement Requirements

1. Private Service Broadcast
2. Mutual Authentication
3. Bilateral Anonymity
4. Bilateral Flexible Policy Control
5. Selective Attribute Disclosure
6. Multi-Show Unlinkability

High Usability Requirements

1. No Pre-registered Pairing
2. No Third-party Dependency during Service Discovery
3. No In-advance Identity Issuance

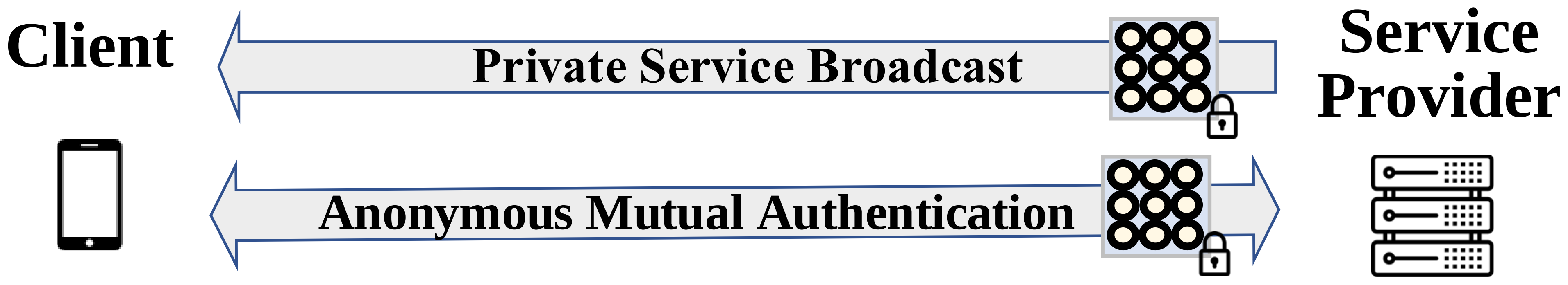
PriSrv: Contributions

- **PriSrv: the first service discovery protocol, to meet both privacy enhancement and high usability requirements**
  - **Core Components of PriSrv**
    - Anonymous Credential-based Matchmaking Encryption (ACME)
    - Fast Anonymous Credential (FAC)
  - **Interoperability with Existing Protocols**
    - Extensible Authentication Protocol (EAP), mDNS, BLE, AirDrop
  - **Deployment on Multiple Platforms in Real Networks**
    - Multiple hardware platforms: desktop, laptop, mobile phone and Raspberry Pi
    - **"immediate response"**: delay stays well bellow 1 second

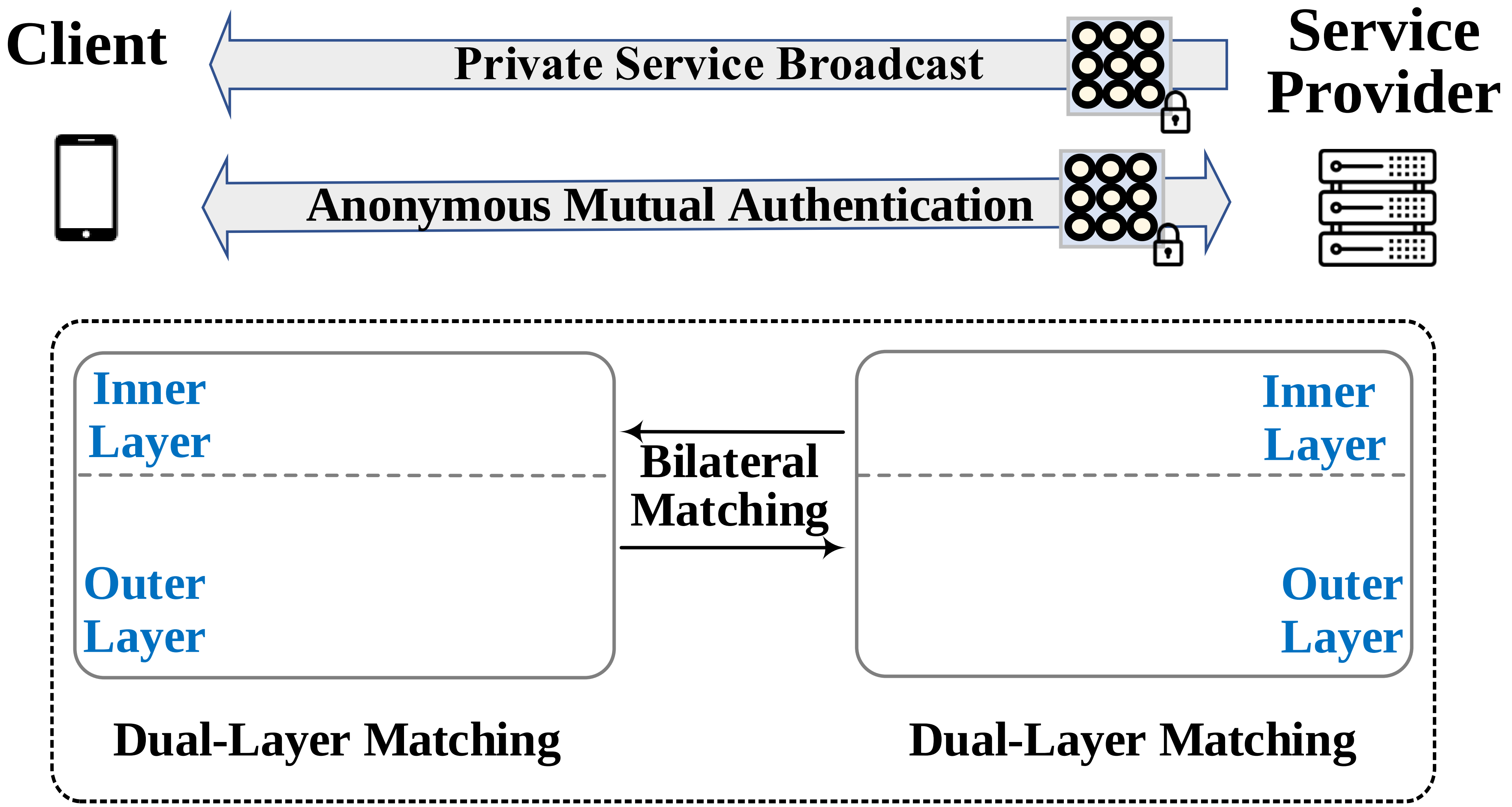
Comparison of SDPs

SD Protocols	Privacy Enhancement						High Usability		
	Private Broadcast	Mutual Authn.	Bilateral Anon.	Bilateral Flex. Pol. Ctrl.	Sel. Attr. Disclosure	Multi-Show Unlinkability	No Pre-reg. Pairing	No 3rd-party Dependence	No In-advance ID Issuance
DNS-SD [18]	×	×	×	×	×	×	√	×	×
mDNS [19]	×	×	×	×	×	×	√	√	×
SSDP [20]	×	×	×	×	×	×	√	√	√
UPnP [21]	×	×	×	×	×	×	√	√	√
Wi-Fi [1]	×	√	×	×	×	×	√	√	×
BLE [3]	×	√	×	×	×	×	√	√	√
AirDrop [2]	×	√	×	×	×	×	√	√	×
PrivateDrop [16]	×	√	√	×	×	×	√	√	×
CBN [9]	×	×	×	×	×	×	×	√	×
WTSB [5]	√	√	√	×	×	×	√	√	×
PriSrv	√	√	√	√	√	√	√	√	×

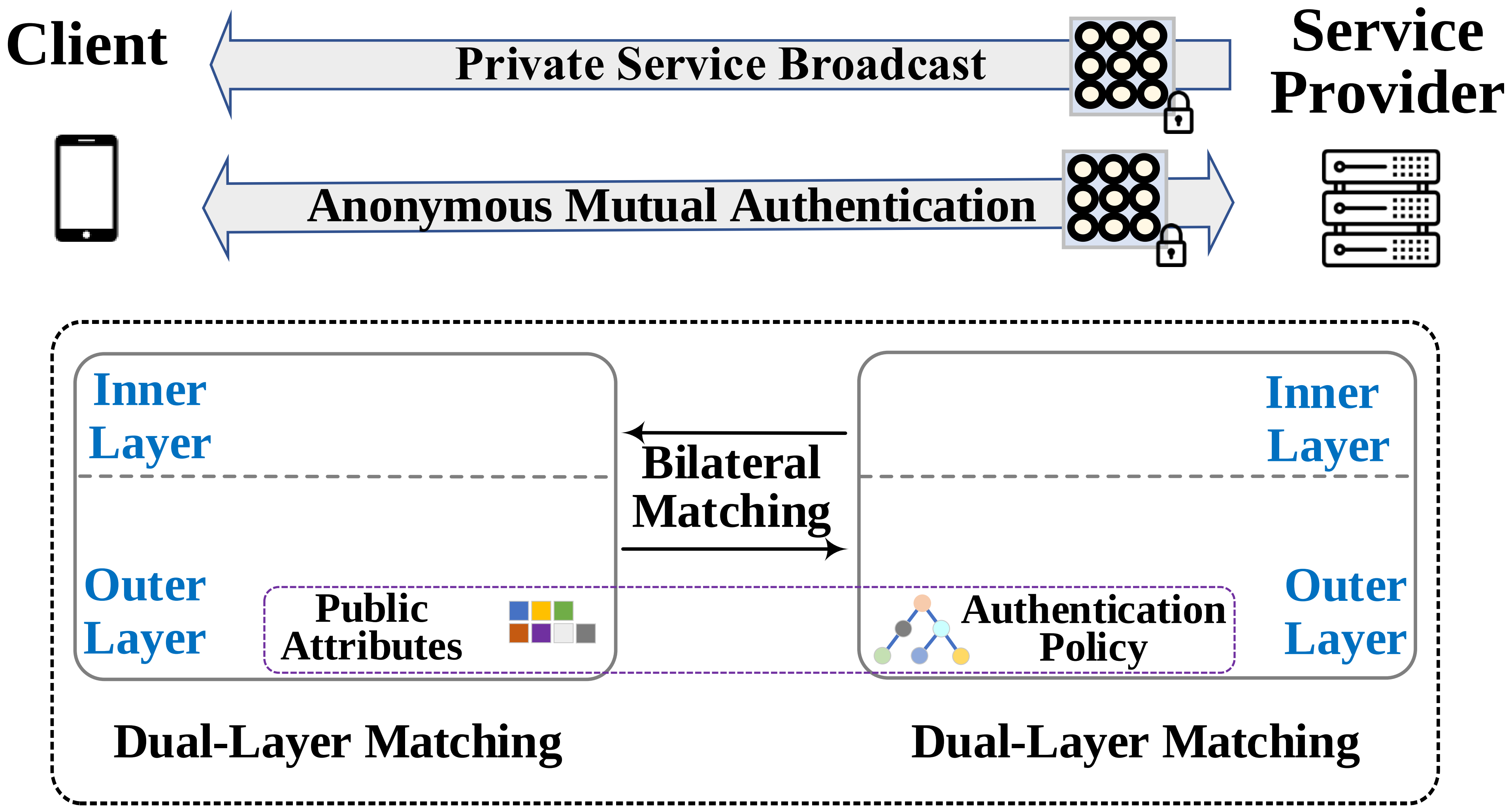
Overview of PriSrv



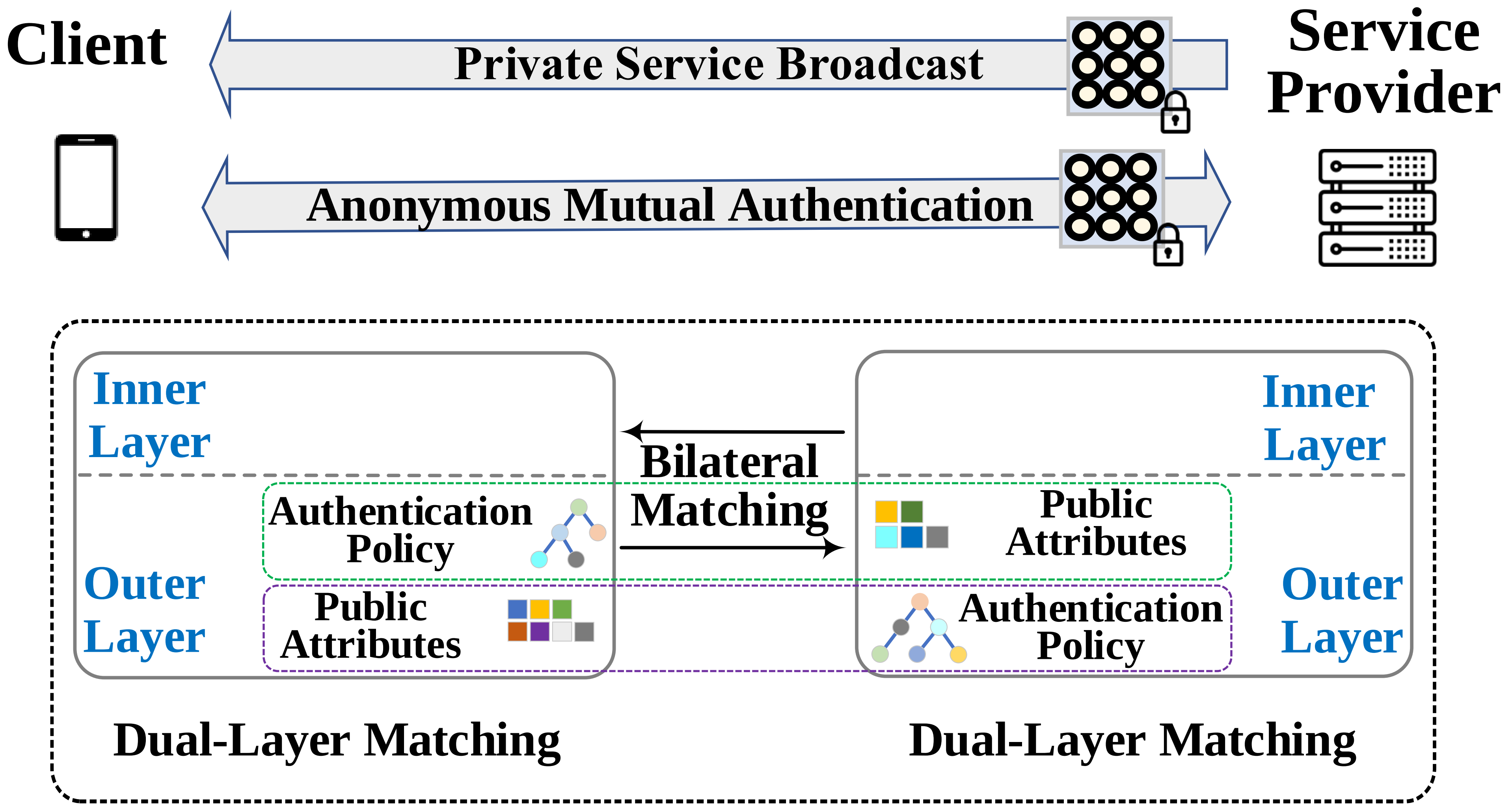
Overview of PriSrv



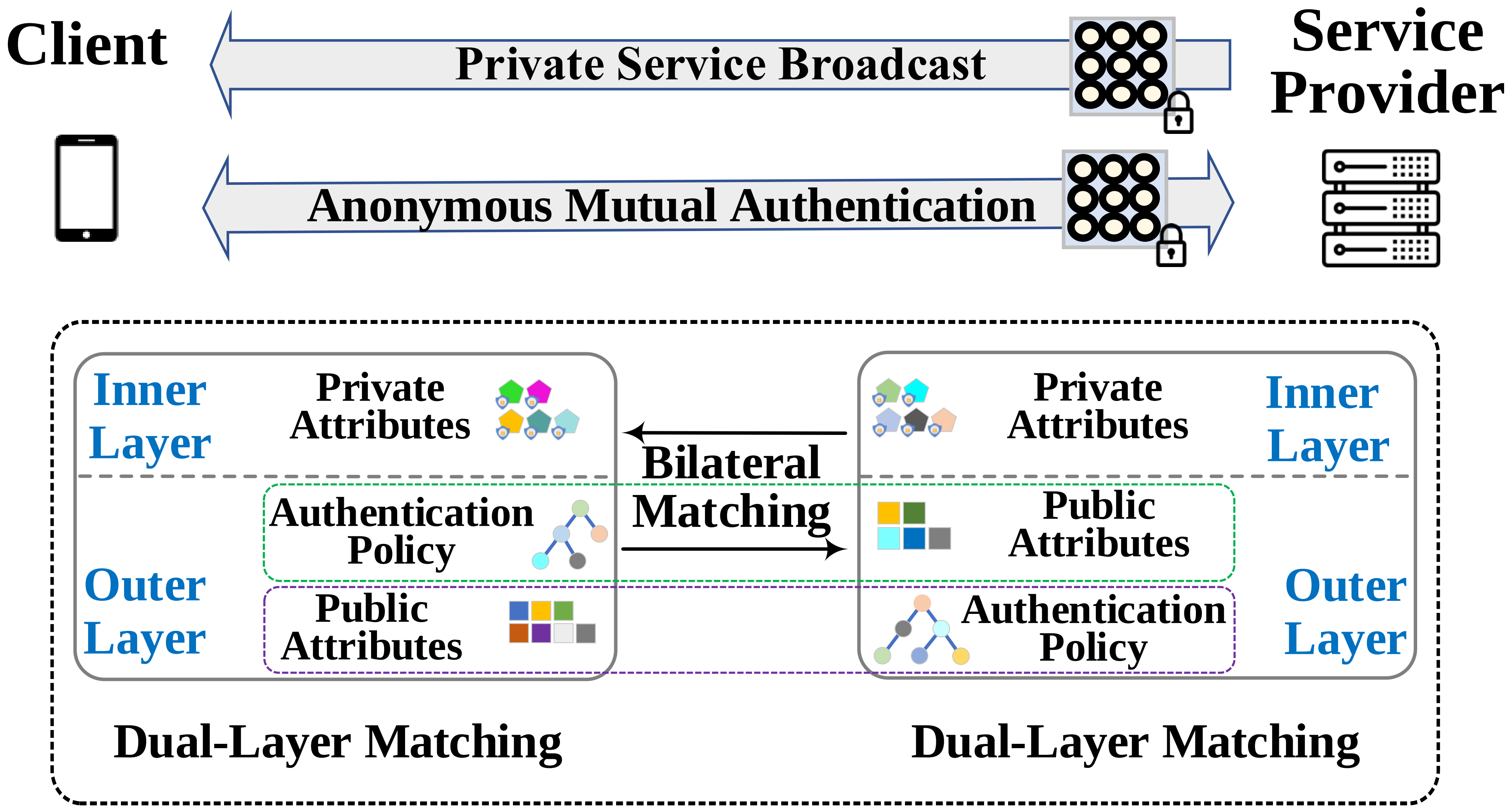
Overview of PriSrv



Overview of PriSrv



Overview of PriSrv



Example

Smart Office: screen mirroring service

Smart TV: Service Provider

Client Device

Example

Smart Office: screen mirroring service

Smart TV: Service Provider

Public Attributes: (device type, vendor, model, OS, domain name)
Private Attributes: (device name, location, IP address, security domain)
Service Policy: Device Type = "Smart phone ∨ Laptop"
∧ OS = "Android ∨ iOS ∨ Windows"
∧ Department = "A ∨ B"

Public Attributes: (device type, model, OS, department)
Private Attributes: (device name, classified device, IP address, security domain)
Connection Policy: Device Type = "TV" ∧ Vendor = "C ∨ D"
∧ Domain Name = "*.XYZ.COM"

Client Device

Example

Smart Office: screen mirroring service

Smart TV: Service Provider

Public Attributes: (device type, vendor, model, OS, domain name)
Private Attributes: (device name, location, IP address, security domain)
Service Policy: Device Type = “Smart phone ∨ Laptop”
∧ OS = “Android ∨ iOS ∨ Windows”
∧ Department = “A ∨ B”

If the **public attributes** of the **service provider** satisfy the **policy** of **client**
and
the **public attributes** of the **client** satisfy the **policy** of **service provider**,

the screen mirroring service can be **discovered** and used by the client.

Public Attributes: (device type, model, OS, department)
Private Attributes: (device name, classified device, IP address, security domain)
Connection Policy: Device Type = “TV” ∧ Vendor = “C ∨ D”
∧ Domain Name = “*.XYZ.COM”

Client Device

Example

Smart Office: screen mirroring service

Smart TV: Service Provider

Public Attributes: (device type, vendor, model, OS, domain name)
Private Attributes: (device name, location, IP address, security domain)
Service Policy: Device Type = “Smart phone ∨ Laptop”
∧ OS = “Android ∨ iOS ∨ Windows”
∧ Department = “A ∨ B”

If the **public attributes** of the **service provider** satisfy the **policy** of **client**
and
the **public attributes** of the **client** satisfy the **policy** of **service provider**,

the screen mirroring service can be **discovered** and used by the client.

The **private attributes** of smart TV and client device are used for **mutual authentication**.

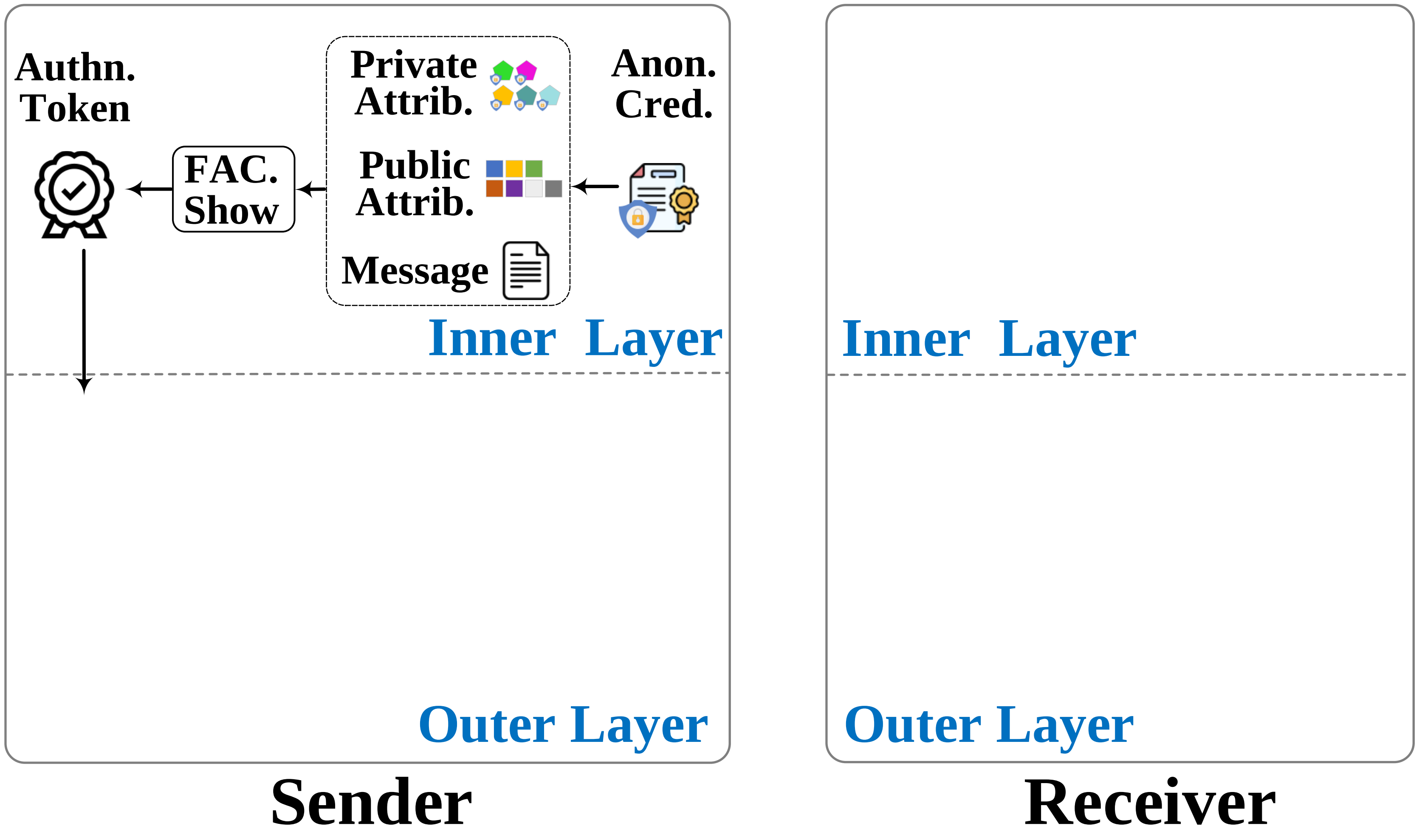
Public Attributes: (device type, model, OS, department)
Private Attributes: (device name, classified device, IP address, security domain)
Connection Policy: Device Type = “TV” ∧ Vendor = “C ∨ D”
∧ Domain Name = “*.XYZ.COM”

Client Device

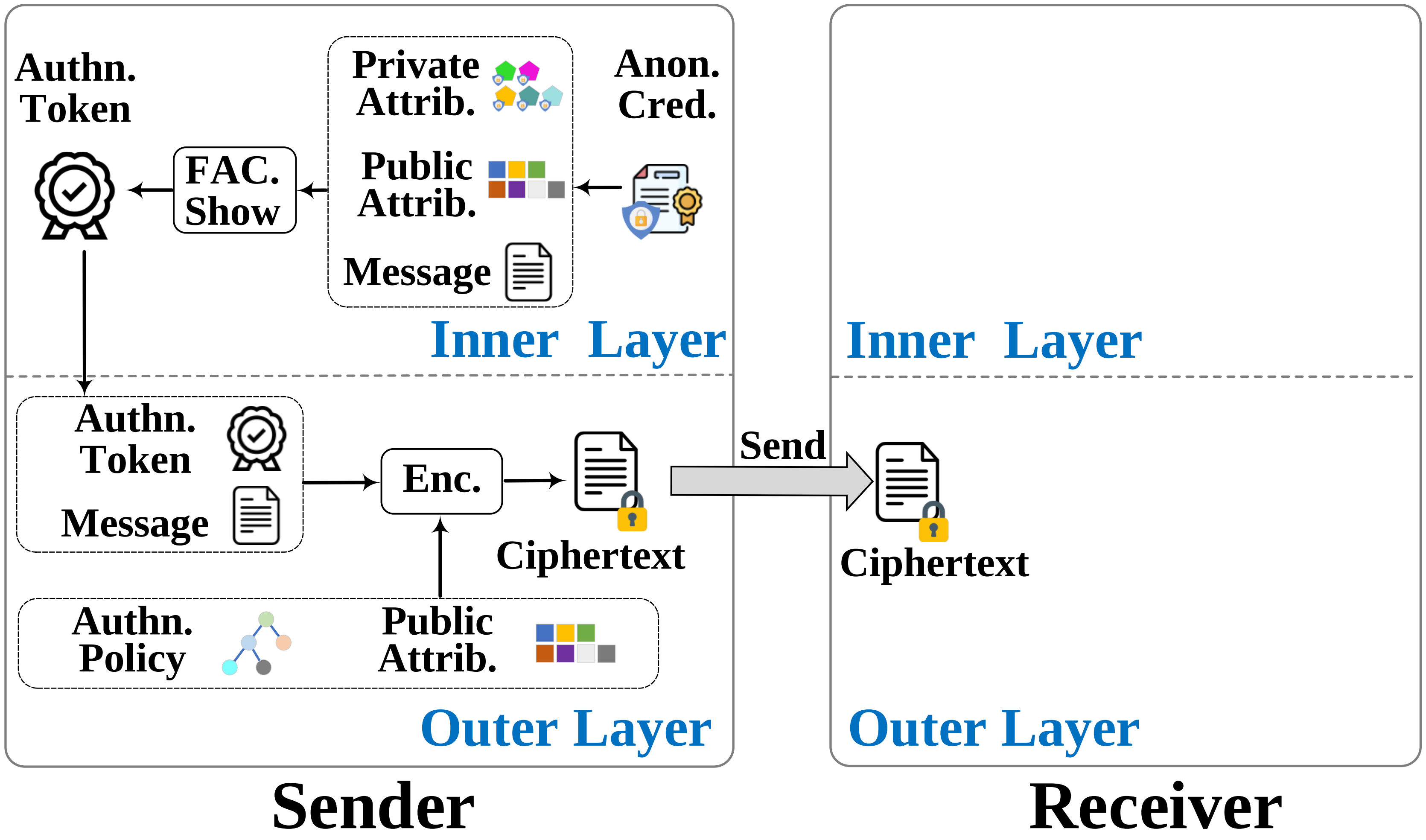
ACME

- Anonymous Credential-based Matchmaking Encryption (ACME)
  - A new cryptographic primitive to support several core features in PriSrv
    - bilateral policy control, anonymous authentication, selective attribute disclosure
  - ACME is a variant of Matchmaking Encryption (ME)
    - The sender and receiver can use anonymous credentials to prove their attributes without revealing their identities
    - Provide stronger privacy guarantees and flexible policy enforcement
  - Fast Anonymous Credential (FAC): Building block of ACME
    - Enable fast anonymous authentication
    - Maintain a constant and small credential size

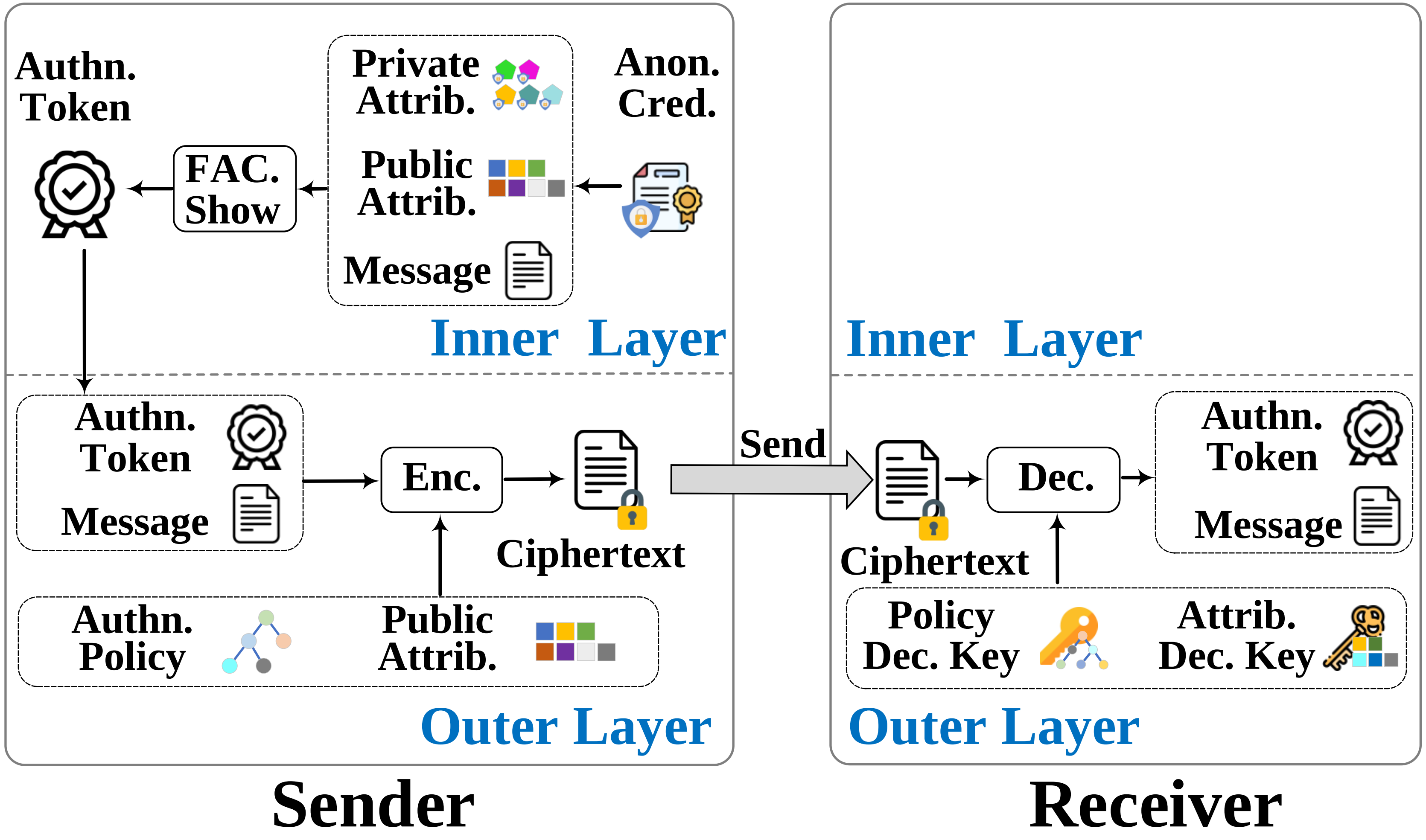
Architecture of ACME



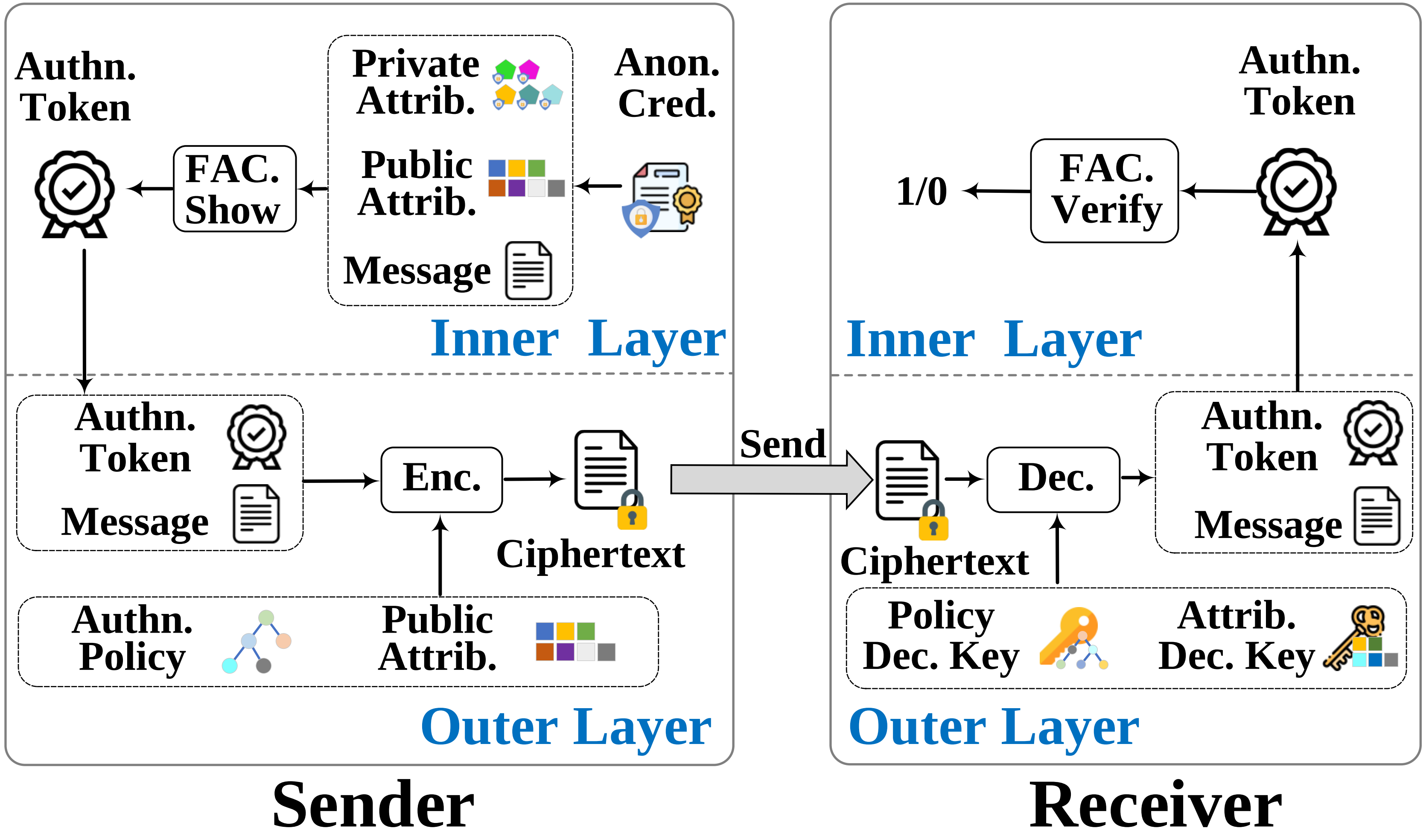
Architecture of ACME



Architecture of ACME



Architecture of ACME



PriSrv Protocol

Service Broadcast Phase
Service Provider S's Broadcast: $bid, \mathsf{CT}_B \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_s, \vec{x}_s, f_s, MSG_B)$
where $MSG_B = \{bid \| Z \| Service_{Type} \| Service_{Par} \| K_c\}$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z \leftarrow h^z \in G_2$, $K_c \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

Anonymous Mutual Authentication Phase	
Client (C)	**Service Provider** (S)
$(\mathsf{cred}_c, \mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c})$	$(\mathsf{cred}_s, \mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s})$

PriSrv Protocol

Service Broadcast Phase

Service Provider S's Broadcast: $bid, \mathsf{CT}_B \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_s, \vec{x}_s, f_s, MSG_B)$

where $MSG_B = \{bid||Z||Service_{Type}||Service_{Par}||K_c\}$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z \leftarrow h^z \in G_2$, $K_c \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

Anonymous Mutual Authentication Phase

Client (C) $(\mathsf{cred}_c, \mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c})$		**Service Provider** (S) $(\mathsf{cred}_s, \mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s})$
$MSG_B \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c}, \mathsf{CT}_B)$		
$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $X_1 \leftarrow g^{x_1} \in G_1$, $X_2 \leftarrow h^{x_2} \in G_2$		
$\sigma_c \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_c, M_c)$		
where $M_c = (\text{“}C \rightarrow S\text{”}, bid, sid, X_1, X_2, Z)$		
$K_s \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$	$\xrightarrow{bid, sid, \sigma_c, \mathsf{CT}_c}$	
$\mathsf{CT}_c \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_c, \vec{x}_c, f_c, MSG_c)$		
where $MSG_c = (K_s, M_c)$		

PriSrv Protocol

Service Broadcast Phase

Service Provider S's Broadcast: $bid, \mathsf{CT}_B \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_s, \vec{x}_s, f_s, MSG_B)$

where $MSG_B = \{bid||Z||Service_{Type}||Service_{Par}||K_c\}$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z \leftarrow h^z \in G_2$, $K_c \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

Anonymous Mutual Authentication Phase

Client (C) $(\mathsf{cred}_c, \mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c})$

Service Provider (S) $(\mathsf{cred}_s, \mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s})$

Client:

$MSG_B \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c}, \mathsf{CT}_B)$

$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $X_1 \leftarrow g^{x_1} \in G_1$, $X_2 \leftarrow h^{x_2} \in G_2$

$\sigma_c \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_c, M_c)$

where $M_c = (\text{“}C \rightarrow S\text{”}, bid, sid, X_1, X_2, Z)$

$K_s \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

$\mathsf{CT}_c \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_c, \vec{x}_c, f_c, MSG_c)$

where $MSG_c = (K_s, M_c)$

Client → Service Provider: $bid, sid, \sigma_c, \mathsf{CT}_c$

Service Provider:

$MSG_c \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s}, \mathsf{CT}_c)$

$b_c \leftarrow \mathcal{MAC}.\mathsf{Verify}(K_c, M_c, \sigma_c)$

If $b_c = 0$, abort; otherwise,

$y \xleftarrow{\$} \mathbb{Z}_p^*$, $Y \leftarrow g^y \in G_1$

$\sigma_s \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_s, M_s)$

where $M_s = (\text{“}S \rightarrow C\text{”}, bid, sid, X_1, X_2, Y, Z)$

$SSK_{c,s} \leftarrow H(X_1^y, X_2^z)$

Service Provider → Client: M_s, σ_s

PriSrv Protocol

Service Broadcast Phase

Service Provider S's Broadcast: $bid, \mathsf{CT}_B \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_s, \vec{x}_s, f_s, MSG_B)$

where $MSG_B = \{bid||Z||Service_{Type}||Service_{Par}||K_c\}$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z \leftarrow h^z \in G_2$, $K_c \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

Anonymous Mutual Authentication Phase

Client (C) $(\mathsf{cred}_c, \mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c})$		**Service Provider** (S) $(\mathsf{cred}_s, \mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s})$
$MSG_B \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c}, \mathsf{CT}_B)$		
$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $X_1 \leftarrow g^{x_1} \in G_1$, $X_2 \leftarrow h^{x_2} \in G_2$		
$\sigma_c \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_c, M_c)$		
where $M_c = (\text{“}C \rightarrow S\text{”}, bid, sid, X_1, X_2, Z)$		
$K_s \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$	$\xrightarrow{bid, sid, \sigma_c, \mathsf{CT}_c}$	$MSG_c \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s}, \mathsf{CT}_c)$
$\mathsf{CT}_c \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_c, \vec{x}_c, f_c, MSG_c)$		$b_c \leftarrow \mathcal{MAC}.\mathsf{Verify}(K_c, M_c, \sigma_c)$
where $MSG_c = (K_s, M_c)$		If $b_c = 0$, abort; otherwise,
		$y \xleftarrow{\$} \mathbb{Z}_p^*$, $Y \leftarrow g^y \in G_1$
$b_s \leftarrow \mathcal{MAC}.\mathsf{Verify}(K_s, M_s, \sigma_s)$		$\sigma_s \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_s, M_s)$
If $b_s = 0$, abort; otherwise,	$\xleftarrow{M_s, \sigma_s}$	where $M_s = (\text{“}S \rightarrow C\text{”}, bid, sid, X_1, X_2, Y, Z)$
$SSK_{c,s} \leftarrow H(Y^{x_1}, Z^{x_2})$		$SSK_{c,s} \leftarrow H(X_1^y, X_2^z)$

PriSrv Protocol

Service Broadcast Phase

Service Provider S's Broadcast: $bid, \mathsf{CT}_B \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_s, \vec{x}_s, f_s, MSG_B)$

where $MSG_B = \{bid||Z||Service_{Type}||Service_{Par}||K_c\}$, $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z \leftarrow h^z \in G_2$, $K_c \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$

Anonymous Mutual Authentication Phase

Client (C) $(\mathsf{cred}_c, \mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c})$		**Service Provider** (S) $(\mathsf{cred}_s, \mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s})$
$MSG_B \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_c}, \mathsf{DK}_{f_c}, \mathsf{CT}_B)$		
$x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $X_1 \leftarrow g^{x_1} \in G_1$, $X_2 \leftarrow h^{x_2} \in G_2$		
$\sigma_c \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_c, M_c)$		
where $M_c = (\text{“}C \rightarrow S\text{”}, bid, sid, X_1, X_2, Z)$		
$K_s \leftarrow \mathcal{MAC}.\mathsf{KeyGen}(1^\lambda)$	$bid, sid, \sigma_c, \mathsf{CT}_c$ $\longrightarrow$	$MSG_c \leftarrow \mathcal{ACME}.\mathsf{Dec}(\mathsf{DK}_{\vec{x}_s}, \mathsf{DK}_{f_s}, \mathsf{CT}_c)$
$\mathsf{CT}_c \leftarrow \mathcal{ACME}.\mathsf{Enc}(\mathsf{cred}_c, \vec{x}_c, f_c, MSG_c)$		$b_c \leftarrow \mathcal{MAC}.\mathsf{Verify}(K_c, M_c, \sigma_c)$
where $MSG_c = (K_s, M_c)$		If $b_c = 0$, abort; otherwise,
		$y \xleftarrow{\$} \mathbb{Z}_p^*$, $Y \leftarrow g^y \in G_1$
$b_s \leftarrow \mathcal{MAC}.\mathsf{Verify}(K_s, M_s, \sigma_s)$		$\sigma_s \leftarrow \mathcal{MAC}.\mathsf{MAC}(K_s, M_s)$
If $b_s = 0$, abort; otherwise,	$\longleftarrow$ M_s, σ_s	where $M_s = (\text{“}S \rightarrow C\text{”}, bid, sid, X_1, X_2, Y, Z)$
$SSK_{c,s} \leftarrow H(Y^{x_1}, Z^{x_2})$		$SSK_{c,s} \leftarrow H(X_1^y, X_2^z)$

Interoperability

- Privacy Enhanced EAP
  - Extends RFC 3748 on Extensible Authentication Protocol (EAP) to support private service discovery

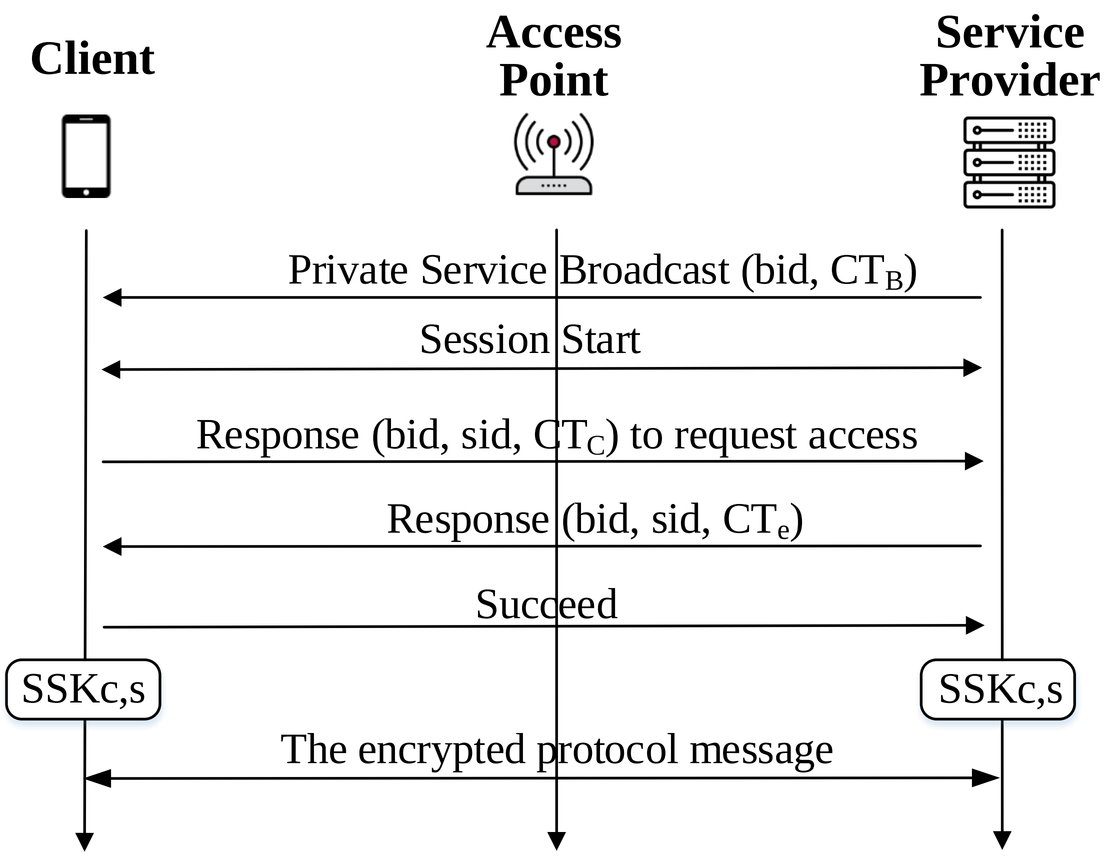


Fig. 4: Architecture of Privacy Enahnced EAP

Interoperability

- Privacy Enhanced EAP
  - Extends RFC 3748 on Extensible Authentication Protocol (EAP) to support private service discovery
- Privacy Enhanced mDNS and BLE
  - PriSrv can be integrated in the Vanadium framework for developing privacy enhanced mDNS and BLE

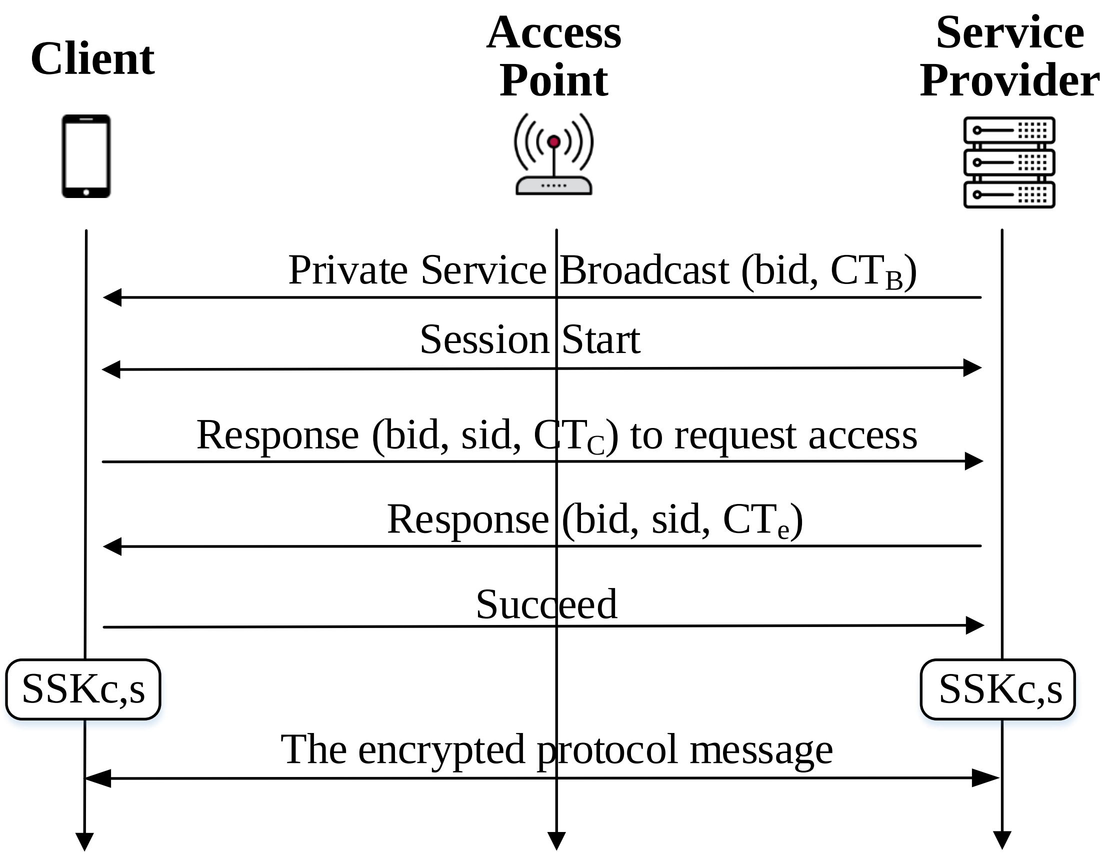


Fig. 4: Architecture of Privacy Enahnced EAP

Interoperability

- Privacy Enhanced EAP
  - Extends RFC 3748 on Extensible Authentication Protocol (EAP) to support private service discovery
- Privacy Enhanced mDNS and BLE
  - PriSrv can be integrated in the Vanadium framework for developing privacy enhanced mDNS and BLE
- Privacy Enhanced Apple AirDrop
  - Avoid transmitting identifier of service provider during broadcast phase
  - Encrypt certificates of both parties using ACME

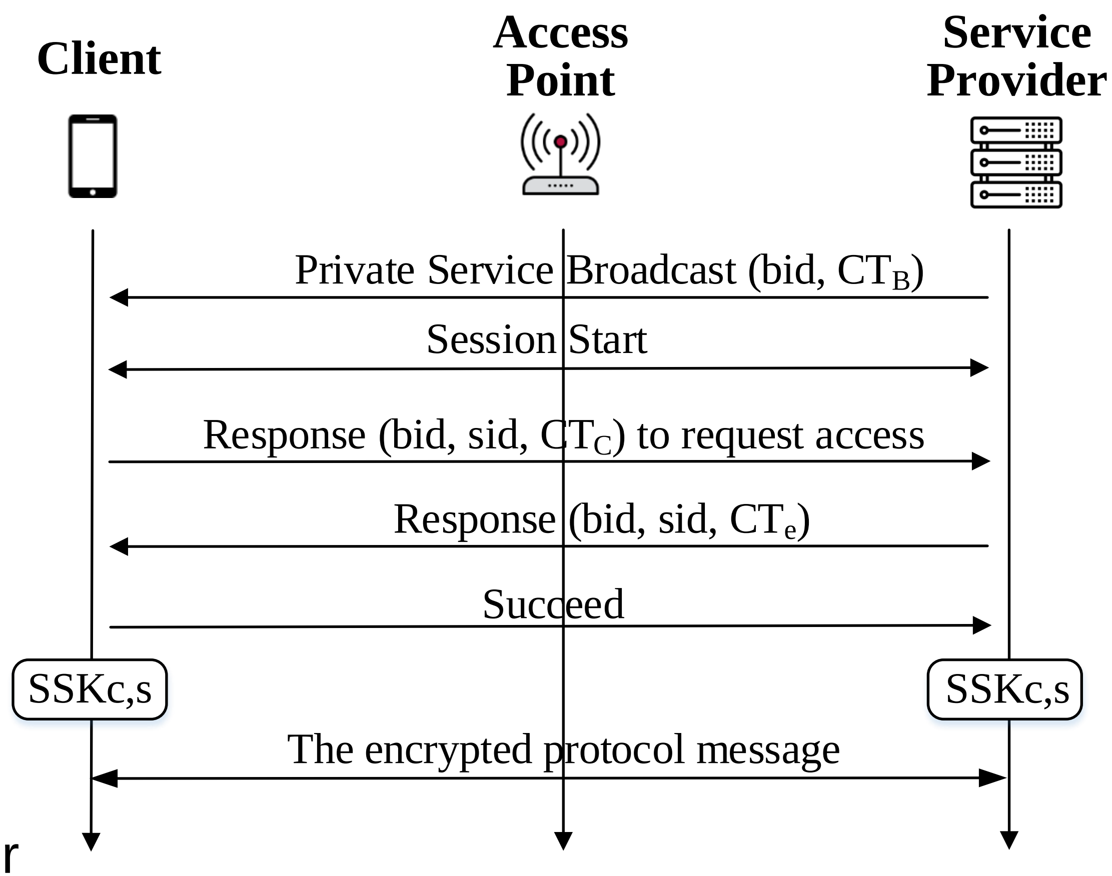


Fig. 4: Architecture of Privacy Enahnced EAP

Implementation and Performance

Device	Private Service Broadcast					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	158.931	164.34	180.337	212.96	202.822	537.98
2	216.493	164.34	261.059	212.96	287.287	537.98
3	385.553	164.34	443.686	212.96	482.725	537.98
4	638.259	164.34	880.868	212.96	1188.392	537.98

Device	Anonymous Mutual Authentication					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	429.282	164.45	517.512	213.09	673.039	538.83
2	576.161	164.45	686.054	213.09	854.177	538.83
3	727.572	164.45	892.712	213.09	972.163	538.83
4	1224.365	164.45	1832.187	213.09	2711.013	538.83

TABLE VII: Performance of PriSrv (ms/KB)

No.	Type	Hardware Platforms
1	Desktop	Intel® Core™ i9-7920X CPU @ 2.9GHz×12, 16GB
2	Laptop	Intel® Core™ i5-10210U CPU @ 1.6GHz×4, 8GB
3	Phone	ARM Cortex @2.84GHz+3×2.4GHz, 4GB
4	Raspberry Pi	ARM Cortex @1.5GHz×4, 2GB

TABLE III: Hardware Platforms for Experiments

Implementation and Performance

Device	Private Service Broadcast					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	158.931	164.34	180.337	212.96	202.822	537.98
2	216.493	164.34	261.059	212.96	287.287	537.98
3	385.553	164.34	443.686	212.96	482.725	537.98
4	638.259	164.34	880.868	212.96	1188.392	537.98

Device	Anonymous Mutual Authentication					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	429.282	164.45	517.512	213.09	673.039	538.83
2	576.161	164.45	686.054	213.09	854.177	538.83
3	727.572	164.45	892.712	213.09	972.163	538.83
4	1224.365	164.45	1832.187	213.09	2711.013	538.83

TABLE VII: Performance of PriSrv (ms/KB)

No.	Type	Hardware Platforms
1	Desktop	Intel® Core™ i9-7920X CPU @ 2.9GHz×12, 16GB
2	Laptop	Intel® Core™ i5-10210U CPU @ 1.6GHz×4, 8GB
3	Phone	ARM Cortex @2.84GHz+3×2.4GHz, 4GB
4	Raspberry Pi	ARM Cortex @1.5GHz×4, 2GB

TABLE III: Hardware Platforms for Experiments

On Device 1-3
Private Service Broadcast Phase: Below 0.538 s
Mutual Authentication Phase: below 0.893 s

Stay well **below 1 s**, which humans perceive the delays as an **"immediate response"**

Implementation and Performance

Device	Private Service Broadcast					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	158.931	164.34	180.337	212.96	202.822	537.98
2	216.493	164.34	261.059	212.96	287.287	537.98
3	385.553	164.34	443.686	212.96	482.725	537.98
4	638.259	164.34	880.868	212.96	1188.392	537.98

Device	Anonymous Mutual Authentication					
	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
1	429.282	164.45	517.512	213.09	673.039	538.83
2	576.161	164.45	686.054	213.09	854.177	538.83
3	727.572	164.45	892.712	213.09	972.163	538.83
4	1224.365	164.45	1832.187	213.09	2711.013	538.83

TABLE VII: Performance of PriSrv (ms/KB)

No.	Type	Hardware Platforms
1	Desktop	Intel® Core™ i9-7920X CPU @ 2.9GHz×12, 16GB
2	Laptop	Intel® Core™ i5-10210U CPU @ 1.6GHz×4, 8GB
3	Phone	ARM Cortex @2.84GHz+3×2.4GHz, 4GB
4	Raspberry Pi	ARM Cortex @1.5GHz×4, 2GB

TABLE III: Hardware Platforms for Experiments

On Device 1-3

Private Service Broadcast Phase: below 0.538 s
Mutual Authentication Phase: below 0.893 s

Stay well **below 1 s**, which humans perceive the delays as an **"immediate response"**

On Device 4

Private Service Broadcast Phase: below 1.189 s
Mutual Authentication Phase: below 2.712 s

Delays are **longer but not too significant**

Limitations and Open Problems

- Large Message Size
  - This large size of the outer discovery broadcast poses a scalability challenge
    - particularly on slower networks like BLE, resulting in high transmission overhead and reception delays
  - Packet Loss
    - clients must wait for the broadcast ciphertext in the subsequent round to receive full packets, causing additional delays in reception

Limitations and Open Problems

- Large Message Size
  - This large size of the outer discovery broadcast poses a scalability challenge
    - particularly on slower networks like BLE, resulting in high transmission overhead and reception delays
  - Packet Loss
    - clients must wait for the broadcast ciphertext in the subsequent round to receive full packets, causing additional delays in reception

- Unlinkability across Multiple layers
  - PriSrv protects its own payloads for achieving unlinkability **at its positioned layer**
  - As for achieving unlinkability at lower layers, the lower layer headers must be protected using specific anti-tracking mechanisms designed at lower layers

Q&A

Thank You!