PriSrv+: Privacy and Usability-Enhanced Wireless Service Discovery with Fast and Expressive Matchmaking Encryption

(NDSS'26, San Diego, USA)

Yang Yang, Guomin Yang, Yingjiu Li, Pengfei Wu, Rui Shi, Minming Huang, Jian Weng, Hwee Hwa Pang, Robert H. Deng

Yang Yang

Singapore Management University
School of Computing and Information Systems (SCIS)
Email: yyang@smu.edu.sg

Service Discovery: Ubiquitous, But Leaky (Privacy & Security Threats)

- **Service discovery (SD)** powers "nearby" connectivity across modern wireless stacks: *Wi-Fi/mDNS/DNS-SD/UPnP/SSDP*, *BLE advertisements*, and *AirDrop-like workflows*.
- **Core design flaw**: SD often relies on **cleartext broadcast advertisements + weak/no authentication**.
  - Broadcast is observable by anyone nearby
  - messages are repeated/structured/stable ⇒ easy to correlate.
- **Threats enabled at scale** (observe *and* inject):
  - **Privacy:** tracking & presence detection, session linkability, identity/service exposure ⇒ profiling.
  - **Security:** spoofing (fake service), MitM (steer to attacker endpoints), DoS (flood discovery channel / disrupt availability).

Note: SD is always-on and broadcast-style, so small leaks become persistent "radio fingerprints".

SMU SINGAPORE MANAGEMENT UNIVERSITY

Required Properties for Private SD

What a "Private SD" Must Provide

- **Policy-controlled discovery + strong privacy + authentication:**
  - Bilateral access control: both sides enforce who can discover/connect
  - Bilateral anonymity: hide identities unless policy grants access
  - Attribute hiding: reveal minimum (e.g., attribute names), keep values private
  - Sender authentication: prevent service spoofing/impersonation
  - Deployability: lightweight enough for BLE/mDNS and constrained devices

Note. Many "private discovery" attempts solve only tracking, but without policy control and authentication, spoofing and unwanted discovery remain.

SMU SINGAPORE MANAGEMENT UNIVERSITY

PriSrv (NDSS'24): Right Architecture, But ACME Bottlenecks

PriSrv[1] recap (what it got right)

- Bilateral policy control with mutual authentication
- Defense against MitM, tracking, and profiling in discovery

Why PriSrv hits limits (inherited from its ACME)

(ACME: Anonymous Credential-based Matchmaking Encryption)

- **Privacy leakage**
  - Outer-layer public attributes may leak ⇒ tracking surface remains
- **Expressiveness & scalability**
  - Binary attribute vectors (0/1) restrict representation and inflate cost for rich policies
  - Small-universe design: fixed attribute set ⇒ adding new attributes is painful
- **Performance & operations**
  - Large ciphertexts ⇒ high communication overhead (heavy for BLE/mDNS)
  - Dependence on pre-issued anonymous credentials ⇒ issuance/revocation management overhead

[1] 1, "PriSrv: Privacy-Enhanced and Highly Usable Service Discovery in Wireless Communications", 2024.

SMU SINGAPORE MANAGEMENT UNIVERSITY

PriSrv+: FEME as the New Core (Fast, Expressive, Deployable)

Key idea: Replace ACME with FEME

(FEME: Fast & Expressive Matchmaking Encryption)

- Expressive bilateral policies: monotonic Boolean formulas
- Unrestricted attribute universe: attributes are arbitrary strings
- Partially hidden access structure: reveal attribute names, hide values
- Built-in privacy-preserving policy matching without leaking sensitive values

Robustness & privacy mechanisms

- **Double re-randomization + binding**
  - blocks encryption-key extraction attempts
  - prevents ciphertext forgery + component-mixing attacks
  - strengthens privacy under expressive policies
- **Randomness splitting** for practical efficiency and clean security arguments

SMU SINGAPORE MANAGEMENT UNIVERSITY

What PriSrv+ achieves over PriSrv

- No external anonymous credential lifecycle
  (removes credential issuance/revocation burden)
- Full attribute-value hiding during discovery (fixes outer-layer exposure in PriSrv)
- Much smaller broadcasts (up to 87% reduction) $\Rightarrow$ better for low bandwidth
- Faster operations (enc/dec and end-to-end broadcast/auth improvements)

Note: PriSrv+ keeps PriSrv's successful architecture, but makes it scalable and practical by redesigning the cryptographic primitive.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Related Work: From IBME to ACME, and Toward FEME

One-to-many and fuzzy variants.

- Sun et al.[2] and Yang et al.[3] proposed privacy-aware ME (PSME) and certificateless ME (CLME), extending IBME to multi-user settings via identity-based broadcast encryption.
- Wu et al.[4] introduced fuzzy IBME (FBME), enabling decryption when attribute overlap exceeds a threshold; FBME has limited expressiveness and incurs high decryption costs.

ACME and FEME.

- Yang et al.[5] developed ACME, an anonymous credential-based ME scheme with flexible bilateral policy control.
- ACME suffers from large ciphertext size and a small-universe construction requiring binary attribute vectors.
- In contrast, FEME supports monotonic Boolean policies with an unrestricted attribute universe (arbitrary strings as attributes), reduces ciphertext size by 87.33%, and achieves up to 7.62× faster encryption and 6.23× faster decryption.

[2] 2, "Privacy-aware and security-enhanced efficient matchmaking encryption", 2023.

[3] 3, "A lightweight certificateless multi-user matchmaking encryption for mobile devices: Enhancing security and performance", 2023.

[4] 4, "Fuzzy identity-based matchmaking encryption and its application", 2023.

[5] 1, "PriSrv: Privacy-Enhanced and Highly Usable Service Discovery in Wireless Communications", 2024.

Roadmap: From ABE to Matchmaking Encryption

We construct FEME, a fast and expressive matchmaking encryption scheme:

- serves as the cryptographic core of PriSrv+
- independent interest for advancing matchmaking encryption

Goal (Matchmaking Encryption: ME).
Sender S and receiver R each have (attributes, policy).
A ciphertext is decryptable iff:

- R's attributes satisfy S's policy, and
- S's attributes satisfy R's policy

$\Rightarrow$ enabling bilateral access control with privacy-preserving policy matching and user anonymity.

Why ABE is the right foundation

- CP-ABE: ciphertext embeds an access policy; key carries attributes
- KP-ABE: key embeds an access policy; ciphertext carries attributes

Both are needed to express the two-sided (sender+receiver) constraints in matchmaking encryption (ME).

SMU SINGAPORE MANAGEMENT UNIVERSITY

Three-Stage Construction

Roadmap: FABEO $\overset{1}{\Longrightarrow}$ Anonymous ABE $\overset{2}{\Longrightarrow}$ Hybrid-ABE $\overset{3}{\Longrightarrow}$ FEME

- **Stage 1: Anonymous Dual-ABE (privacy + efficiency)**
  Build A-CP-ABE and A-KP-ABE from FABEO:
  - Hide attribute values in: access policies (CP side); attribute sets (KP side)
  - Reduce expensive operations tied to policy size
    $\Rightarrow$ improves both privacy and runtime for expressive policies.
- **Stage 2: Hybrid-ABE (bridge Dual-ABE $\rightarrow$ ME, adds sender authentication)**
  Introduce Hybrid-ABE to support:
  - Bilateral policy matching (connect CP-style and KP-style checks)
  - Sender authentication inside the cryptographic flow
    $\Rightarrow$ prevents "who encrypted this?" ambiguity and blocks abuse in SD settings.
- **Stage 3: FEME (full ME with privacy-preserving matching + anonymity)**
  Integrate A-CP-ABE + A-KP-ABE + Hybrid-ABE into FEME

SMU SINGAPORE MANAGEMENT UNIVERSITY

Stage 1: Randomness Splitting to Stop Attribute Guessing

Problem in original FABEO KP-ABE

- Reuse of single randomness s creates a "test handle": $\mathrm{ct}_{1,u} = H(u)^s$, $\mathrm{ct}_2 = g_2^s$
- Attacker guesses u and checks: $e(\mathrm{ct}_{1,u}, g_2) \stackrel{?}{=} e(H(u), \mathrm{ct}_2)$
  $\Rightarrow$ offline attribute guessing.

Fix: split randomness

- Sample s', s'' and set $s = s' + s''$
- Modify ciphertext anchors: $\mathrm{ct}_{1,i} = H(u_i)^s$, $\mathrm{ct}_2 = \delta_1^{s'}$, $\mathrm{ct}_3 = \delta_2^{s''}$, where $\delta_1 = g_2^{b_1}$, $\delta_2 = g_2^{b_2}$.

Key-side cancellation

- Decryption keys include components using exponents $1/b_1$ and $1/b_2$
  $\Rightarrow$ legitimate decrypt cancels b_1, b_2; attacker cannot test guesses.

Result: ciphertext reveals no usable equality test $\Rightarrow$ guessing blocked.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Stage 1: CP-ABE Bottleneck Removal (Vector $\rightarrow$ Scalar)

FABEO CP-ABE bottleneck

Ciphertext uses vector randomness $\vec{s}' \in \mathbb{Z}_p^\tau \Rightarrow$ decryption cost scales with τ:

- τ pairings; about τl exponentiations (where $l = \#$attributes actually used to satisfy the policy)

Optimization

- Replace vector randomness with scalar: $s' \leftarrow_\$ \mathbb{Z}_p$
- Simplify ciphertext components: $\mathsf{ct}_2 = g_2^{s'}$, $\mathsf{ct}_{3,i} = h^{\mathbf{M}_i(s_1 || \mathbf{v})^\top} \cdot H(\Psi_{\pi(i)})^{s'}$.

New decryption cost

- 1 pairing + l exponentiations
  $\Rightarrow$ major speedup for expressive policies.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Stage 2: Hybrid-ABE (Bridge ABE ⇒ ME via Sender Authentication)

Goal of stage 2

- ME requires sender authentication: only authorized senders (with valid EK tied to their attributes) can produce valid ciphertexts.
- Plain CP-ABE/KP-ABE encryption uses only mpk+ policy/attributes ⇒ **no sender binding**.

Hybrid-ABE design

- EKGen: produces sender encryption key $\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}$ (CP-ABE-style keygen + KP-ABE-style randomness splitting)
- PolGen: produces receiver policy key $\mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}$ (KP-ABE exponentiation tricks)

Encryption intuition

- Ciphertext wraps message with: $ct_0 = Y^s \cdot \mathsf{msg}$
- Re-randomize the sender key into ciphertext components using nonce τ' and split randomness $s = s' + s''$ ⇒ ciphertext is bound to a legitimate sender key

Efficiency target

- Decryption unified across the two sides with small constant pairing count and $O(I)$ exponentiations.

Hybrid-ABE: Bridging CP-ABE and KP-ABE

$\underline{\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})}$. Generate $\mathcal{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. Pick $x, \mu, b_1, b_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $h \xleftarrow{\$} \mathbb{G}_1$ and a hash function $H : \{0,1\}^* \to \mathbb{G}_1$. Compute $Y = e(g_1, g_2)^{x\mu}$, $\delta_0 = g_2^{\mu}$, $\delta_1 = g_2^{b_1}$, $\delta_2 = g_2^{b_2}$. Output the master public key $\mathsf{mpk} := (\mathcal{G}, H, Y, h, \delta_0, \delta_1, \delta_2)$ and master secret key $\mathsf{msk} := (x, \mu, b_1, b_2)$.

$\underline{\mathsf{EKGen}(\mathsf{msk}, \mathcal{S}_{\mathsf{snd}} = \{u_i\}_{i\in[\ell]} = \{\langle n_i, v_i\rangle\}_{i\in[\ell]}) \to \mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}}$. Pick $\tau \xleftarrow{\$} \mathbb{Z}_p^*$. Compute $\mathsf{ek}_1 = g_1^x h^\tau$, $\mathsf{ek}_3 = \delta_1^\tau$, $\mathsf{ek}_4 = \delta_2^\tau$, $\mathsf{ek}_{2,i} = H(u_i)^\tau$. Output $\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}} := (\{n_i\}_{i\in[\ell]}, \mathsf{ek}_1, \{\mathsf{ek}_{2,i}\}_{i\in\ell}, \mathsf{ek}_3, \mathsf{ek}_4)$.

$\underline{\mathsf{PolGen}(\mathbb{A}_{\mathsf{rcv}} = (\mathsf{msk}, \mathbf{A}, \rho, \{\Psi_{\rho(i)}\}_{i\in[m]})) \to \mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}}$. Pick $r' \xleftarrow{\$} \mathbb{Z}_p^*$ and $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_p^{n-1}$. Compute $\mathsf{sk}_1 = g_2^{r'}$, $\mathsf{sk}_{2,i} = \left(h^{\mathbf{M}_i(s_1||\mathbf{y})^\top} \cdot H(\Psi_{\rho(i)})^{r'}\right)^{\frac{1}{b_1}}$, $\mathsf{sk}_{3,i} = \left(h^{\mathbf{M}_i(s_1||\mathbf{y})^\top} \cdot H(\Psi_{\rho(i)})^{r'}\right)^{\frac{1}{b_2}}$. Output $\mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}} := ((\mathbf{A}, \rho, \{n_{\rho(i)}\}_{i\in[m]}), \mathsf{sk}_1, \{\mathsf{sk}_{2,i}, \mathsf{sk}_{3,i}\}_{i\in[m]})$.

$\underline{\mathsf{Enc}(\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}, \mathsf{msg}) \to \mathsf{CT}_{\mathsf{snd}}}$. Pick $\tau', s', s'' \xleftarrow{\$} \mathbb{Z}_p^*$. Let $s = s' + s''$. Compute $\mathsf{ct}_0 = Y^s \cdot \mathsf{msg}$, $\mathsf{ct}_{1,i} = (\mathsf{ek}_{1,i} \cdot H(u_i)^{\tau'})^s$, $\mathsf{ct}_2 = (\mathsf{ek}_2 \cdot \delta_1^{\tau'})^{s'}$, $\mathsf{ct}_3 = (\mathsf{ek}_2 \cdot \delta_2^{\tau'})^{s''}$, $\mathsf{ct}_4 = (\mathsf{ek}_4 \cdot h^{\tau'})^s$. Output $\mathsf{CT}_{\mathsf{snd}} := (\{n_i\}_{i\in[\ell]}, \mathsf{ct}_0, \{\mathsf{ct}_{1,i}\}_{i\in[\ell]}, \mathsf{ct}_2, \mathsf{ct}_3, \mathsf{ct}_4)$.

$\underline{\mathsf{Dec}(\mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}, \mathsf{CT}_{\mathsf{snd}}) \to \mathsf{msg}/\perp}$. If there is any subset I that matches $\{n_i\}_{i\in[\ell]}$ in $\mathsf{CT}_{\mathsf{snd}}$ with $(\mathbf{A}, \rho, \{n_{\rho(i)}\}_{i\in[m_2]})$ in $\mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}$, there exist constants $\{\omega_i\}_{i\in I}$ s.t. $\sum_{i\in I} \omega_i \mathbf{A}_i = (1, 0, \cdots, 0)$. Output

$$\mathsf{msg} = \mathsf{ct}_0 \cdot \frac{e(\prod_{i\in I}(\mathsf{sk}_{2,i})^{\omega_i}, \mathsf{ct}_2)\, e(\prod_{i\in I}(\mathsf{sk}_{3,i})^{\omega_i}, \mathsf{ct}_3)}{e(\mathsf{ct}_4, \delta_0)\, e(\prod_{i\in I}(\mathsf{ct}_{1,\rho(i)})^{\omega_i}, \mathsf{sk}_1)}.$$

AGEMENT

Stage 3: FEME (Full Integration of A-CP-ABE + A-KP-ABE + Hybrid-ABE)

FEME algorithms: Setup, EKGen, DKGen, PolGen, Enc, Dec

Key generation roles

- EKGen: sender attribute encryption key (from Hybrid-ABE)
- DKGen: receiver attribute decryption key (from A-CP-ABE keygen)
- PolGen: receiver policy key combining:
  - A-KP-ABE components $(\mathsf{sk}_1, \{sk_{2,i}, sk_{3,i}\})$
  - Hybrid-ABE-style additional components $(\{sk_{4,i}, sk_{5,i}\})$

Encryption composition (big picture)

- Authenticated encapsulation: $\mathsf{ct}_0 = \phi(\mathsf{msg}) \oplus \hat{H}(V),\ V = Z^{s_1+s_2} \cdot Y^{s_3}$,
- Produce ciphertext blocks by "stacking":
  - A-CP-ABE block: $(\mathsf{ct}_1, ct_2, \{ct_{3,i}\})$
  - A-KP-ABE block: $(\mathsf{ct}_{4,1}, ct_{4,2}, \{ct_{5,i}\})$
  - Hybrid-ABE block: $(\{ct_{6,i}\}, \mathsf{ct}_7, ct_8, ct_9)$

Decryption

- Runs three coupled decrypt fractions (one per block) and combines them to recover V, msg.
  $\Rightarrow$ achieves bilateral policy matching + anonymity + sender authentication with high efficiency.

Stage 3: FEME Scheme

FEME: Fast and Expressive Matchmaking Encryption

1. $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{msk})$. // System Setup

This algorithm takes in the security parameter 1^λ and generates a bilinear pairing $\mathcal{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. The algorithm picks random numbers $\alpha, x, \mu, b_1, b_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $h \xleftarrow{\$} \mathbb{G}_1$, hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}_1$, $\hat{H} : \mathbb{G}_T \rightarrow \{0,1\}^{l_0}$, and a polynomial-time computable padding function $\phi : \{0,1\}^n \rightarrow \{0,1\}^{\ell_0}$. It computes $Z = e(g_1, g_2)^\alpha, Y = e(g_1, g_2)^{x\mu}, \delta_0 = g_2^\mu, \delta_1 = g_2^{b_1}, \delta_2 = g_2^{b_2}$. It outputs the master public key as $\mathsf{mpk} := (\mathcal{G}, H, \hat{H}, \phi, Z, Y, h, \delta_0, \delta_1, \delta_2)$, and the master secret key as $\mathsf{msk} := (\alpha, x, \mu, b_1, b_2)$.

2. $\mathsf{EKGen}(\mathsf{msk}, \mathcal{S}_{\mathsf{snd}} = \{u_i\}_{i\in[\ell_1]} = \{\langle n_i, v_i\rangle\}_{i\in[\ell_1]}) \rightarrow \mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}$. // Attribute Encryption Key Generation

This algorithm generates the sender's attribute encryption key $\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}$ for attributes $\mathcal{S}_{\mathsf{snd}} = \{u_i\}_{i\in[\ell_1]} = \{\langle n_i, v_i\rangle\}_{i\in[\ell_1]}$, where n_i denotes the attribute name and v_i the attribute value. It picks a random number $\tau \xleftarrow{\$} \mathbb{Z}_p^*$ and computes as follows: $\mathsf{ek}_{1,i} = H(u_i)^\tau$ for $i \in [\ell_1]$, $\mathsf{ek}_2 = \delta_1^\tau$, $\mathsf{ek}_3 = \delta_2^\tau$, $\mathsf{ek}_4 = g_1^\alpha h^\tau$. It outputs the sender attribute encryption key $\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}} := (\{n_i\}_{i\in[\ell_1]}, \{\mathsf{ek}_{1,i}\}_{i\in[\ell_1]}, \mathsf{ek}_2, \mathsf{ek}_3, \mathsf{ek}_4)$.

3. $\mathsf{DKGen}(\mathsf{msk}, \mathcal{S}_{\mathsf{rcv}} = \{u_i\}_{i\in[\ell_2]} = \{\langle n_i, v_i\rangle\}_{i\in[\ell_2]}) \rightarrow \mathsf{DK}_{\mathcal{S}_{\mathsf{rcv}}}$. // Attribute Decryption Key Generation

To generate the receiver's attribute decryption key $\mathsf{DK}_{\mathcal{S}_{\mathsf{rcv}}}$ for attributes $\mathcal{S}_{\mathsf{rcv}} = \{u_i\}_{i\in[\ell_2]} = \{\langle n_i, v_i\rangle\}_{i\in[\ell_2]}$, this algorithm picks a random number $r \xleftarrow{\$} \mathbb{Z}_p^*$ and computes as follows: $\mathsf{dk}_1 = g_1^\alpha h^r$, $\mathsf{dk}_{2,i} = H(u_i)^r$, $\mathsf{dk}_3 = g_2^r$. It outputs the receiver attribute decryption key $\mathsf{DK}_{\mathcal{S}_{\mathsf{rcv}}} := (\{n_i\}_{i\in[\ell_2]}, \mathsf{dk}_1, \{\mathsf{dk}_{2,i}\}_{i\in[\ell_2]}, \mathsf{dk}_3)$.

4. $\mathsf{PolGen}(\mathsf{msk}, \mathbb{A}_{\mathsf{rcv}} = (\mathbf{A}, \rho, \{\Psi_{\rho(i)}\}_{i\in[m_2]})) \rightarrow \mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}$. // Policy Decryption Key Generation

This receiver's policy decryption key generation algorithm generates the secret key $\mathsf{sk}_{\mathbb{A}_{\mathsf{rcv}}}$ with receiver's monotone span policy $\mathbb{A}_{\mathsf{rcv}} = (\mathbf{A}, \rho, \{\Psi_{\rho(i)}\}_{i\in[m_2]})$, where $\mathbf{A}$ is an $m_2 \times n_2$ access control matrix, $\{\Psi_{\rho(i)}\}_{i\in[m_2]} = \{\langle n_{\rho(i)}, v_{\rho(i)}\rangle\}_{i\in[m_2]}$, $n_{\rho(i)}$ denotes attribute name and $v_{\rho(i)}$ attribute value. It picks a random number $r' \xleftarrow{\$} \mathbb{Z}_p^*$, a random vector $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_p^{n_2-1}$ and computes as follows:

$$\mathsf{sk}_1 = g_2^{r'}, \quad \mathsf{sk}_{2,i} = (g_1^{\mathbf{A}_i(\alpha||\mathbf{y})^\top} \cdot H(\Psi_{\rho(i)})^{r'})^{\frac{1}{b_1}}, \quad \mathsf{sk}_{3,i} = (g_1^{\mathbf{A}_i(\alpha||\mathbf{y})^\top} \cdot H(\Psi_{\rho(i)})^{r'})^{\frac{1}{b_2}},$$

SMU SINGAPORE MANAGEMENT UNIVERSITY

Stage 3: FEME Scheme (con't)

5. $\mathsf{Enc}(\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}, \mathbb{A}_{\mathsf{snd}} = (\mathbf{M}, \pi, \{\Psi_{\pi(i)}\}_{i\in[m_1]}), \mathsf{msg}) \to \mathsf{CT}_{\mathsf{snd}}$. // Encrypt

This algorithm encrypts a message $\mathsf{msg} \in \{0,1\}^n$ with sender's monotone span policy $\mathbb{A}_{\mathsf{snd}} = (\mathbf{M}, \pi, \{\Psi_{\pi(i)}\}_{i\in[m_1]})$ and sender attribute encryption key $\mathsf{EK}_{\mathcal{S}_{\mathsf{snd}}}$, where $\{\Psi_{\pi(i)}\}_{i\in[m_1]} = \{\langle n_{\pi(i)}, v_{\pi(i)}\rangle\}_{i\in[m_1]}$ and matrix $\mathbf{M} \in \mathbb{Z}_p^{m_1\times n_1}$. It selects $s_1, s_2', s_2'', s_3', s_3'', \tau' \xleftarrow{\$} \mathbb{Z}_p^*$, a vector $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^{n_1-1}$. Let $s_2 = s_2' + s_2''$ and $s_3 = s_3' + s_3''$. It computes as follows:

$$V = Z^{s_1+s_2} \cdot Y^{s_3}, \quad \mathsf{ct}_0 = \phi(\mathsf{msg}) \oplus \hat{H}(V), \quad \mathsf{ct}_1 = g_2^{s_1}, \quad \mathsf{ct}_2 = g_2^{s_3},$$

$$\mathsf{ct}_{3,i} = h^{\mathbf{M}_i(s_1\|\mathbf{v})^\top} \cdot H(\Psi_{\pi(i)})^{s_3} \text{ for each row } i \in [m_1], \quad \mathsf{ct}_{4,1} = \delta_1^{s_2'}, \quad \mathsf{ct}_{4,2} = \delta_2^{s_2''},$$

$$\mathsf{ct}_{5,i} = H(u_i)^{s_2}, \quad \mathsf{ct}_{6,i} = (\mathsf{ek}_{1,i}\cdot H(u_i)^{\tau'})^{s_3} \text{ for } i \in [\ell_1], \quad \mathsf{ct}_7 = (\mathsf{ek}_2\cdot\delta_1^{\tau'})^{s_3'}, \quad \mathsf{ct}_8 = (\mathsf{ek}_3\cdot\delta_2^{\tau'})^{s_3''}, \quad \mathsf{ct}_9 = (\mathsf{ek}_4\cdot h^{\tau'})^{s_3}.$$

It outputs the ciphertext

$$\mathsf{CT}_{\mathsf{snd}} := ((\mathbf{M}, \pi, \{n_{\pi(i)}\}_{i\in[m_1]}), \{n_i\}_{i\in[\ell_1]}, \mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ct}_2, \{\mathsf{ct}_{3,i}\}_{i\in[m_1]}, \mathsf{ct}_{4,1}, \mathsf{ct}_{4,2}, \{\mathsf{ct}_{5,i}, \mathsf{ct}_{6,i}\}_{i\in[\ell_1]}, \mathsf{ct}_7, \mathsf{ct}_8, \mathsf{ct}_9).$$

6. $\mathsf{Dec}(\mathsf{DK}_{\mathcal{S}_{\mathsf{rcv}}}, \mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}, \mathsf{CT}_{\mathsf{snd}}) \to \mathsf{msg}/\perp$: // Decrypt

This algorithm decrypts a given ciphertext $\mathsf{CT}_{\mathsf{snd}}$ using $\mathsf{DK}_{\mathcal{S}_{\mathsf{rcv}}}$ and $\mathsf{SK}_{\mathbb{A}_{\mathsf{rcv}}}$. If $\mathcal{S}_{\mathsf{rcv}} \models \mathbb{A}_{\mathsf{snd}}$ (denoting that $\mathcal{S}_{\mathsf{rcv}}$ satisfies $\mathbb{A}_{\mathsf{snd}}$), there exist constants $\{\gamma_i\}_{i\in I_1}$ s.t. $\sum_{i\in I_1}\gamma_i\mathbf{M}_i = (1,0,\cdots,0)$. If $\mathcal{S}_{\mathsf{snd}} \models \mathbb{A}_{\mathsf{rcv}}$ (denoting that $\mathcal{S}_{\mathsf{snd}}$ satisfies $\mathbb{A}_{\mathsf{rcv}}$), there exist constants $\{\omega_i\}_{i\in I_2}$ s.t. $\sum_{i\in I_2}\omega_i\mathbf{A}_i = (1,0,\cdots,0)$. This algorithm recovers V by computing

$$V = \frac{e(\mathsf{dk}_1, \mathsf{ct}_1)e(\prod_{i\in I_1}(\mathsf{dk}_{2,\pi(i)})^{\gamma_i}, \mathsf{ct}_2)}{e(\prod_{i\in I_1}(\mathsf{ct}_{3,\pi(i)})^{\gamma_i}, \mathsf{dk}_3)} \cdot \frac{e(\prod_{i\in I_2}(\mathsf{sk}_{2,\rho(i)})^{\omega_i}, \mathsf{ct}_{4,1})e(\prod_{i\in I_2}(\mathsf{sk}_{3,\rho(i)})^{\omega_i}, \mathsf{ct}_{4,2})}{e(\prod_{i\in I_2}(\mathsf{ct}_{5,\rho(i)})^{\omega_i}, \mathsf{sk}_1)}$$

$$\cdot \frac{e(\mathsf{ct}_9, \delta_0)e(\prod_{i\in I_2}(\mathsf{ct}_{6,\rho(i)})^{\omega_i}, \mathsf{sk}_1)}{e(\prod_{i\in I_2}(\mathsf{sk}_{4,\rho(i)})^{\omega_i}, \mathsf{ct}_7)e(\prod_{i\in I_2}(\mathsf{sk}_{5,\rho(i)})^{\omega_i}, \mathsf{ct}_8)}.$$

It computes $\phi(\mathsf{msg}) = \mathsf{ct}_0 \oplus \hat{H}(V)$. If the padding is valid, this algorithm returns msg. Otherwise, it returns $\perp$.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Comparative Advantages of FEME (Table)

Scheme	Expressiveness			Security and Privacy			Usability	
	Monotonic Policy	Arbitrary Attribute	Large Universe	Data Privacy	Data Authenticity	Attribute Privacy	No Pre-registration Pairing	No Additional Component
IBME[6] (Crypto'19)	×	×	√	√	√	√	×	√
IBME[7] (IndoCrypt'21)	×	×	√	√	×/√[8]	√	×	√
IBME[9] (AsiaCrypt'21)	×	×	√	√	√	√	×	√
FBME[10] (TIFS'23)	×	×	×	√	√	√	×	√
PSME[11] (TIFS'23)	×	×	×	√	√	√	×	√
CLME[12] (TIFS'23)	×	×	√	√	√	√	×	√
ACME[13] (NDSS'24)	√	×	×	√	√	×	√	×
FEME	√	√	√	√	√	√	√	√

Table 1: Comparison of Matchmaking Encryption (ME) Schemes

[6] S, "Match me if you can: Matchmaking encryption and its applications", 2019.

[7] S, "Identity-Based Matchmaking Encryption Without Random Oracles", 2021.

[8] IBME (IndoCrypt'21) contains two schemes: one with authenticity; and another not.

[9] S, "Identity-based matchmaking encryption from standard assumptions", 2022.

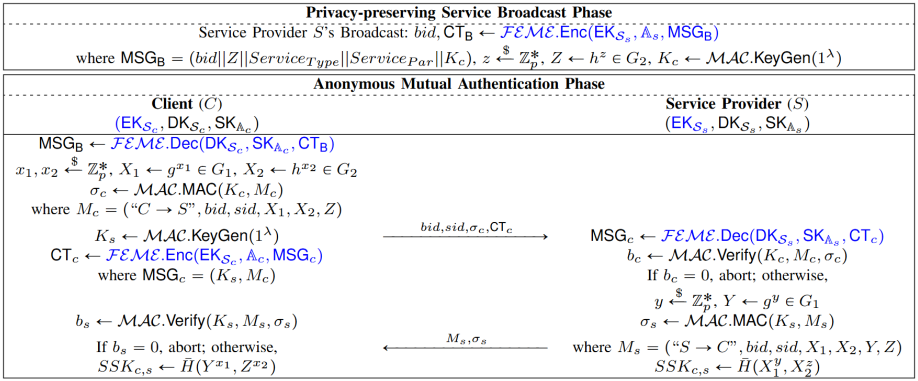
[10] S, "Fuzzy identity-based matchmaking encryption and its application", 2023.

[11] S, "Privacy-aware and security-enhanced efficient matchmaking encryption", 2023.

[12] S, "A lightweight certificateless multi-user matchmaking encryption for mobile devices: Enhancing security and performance", 2023.

[13] S, "PriSrv: Privacy-Enhanced and Highly Usable Service Discovery in Wireless Communications", 2024.

PriSrv+ Protocol



Theorem.

- Suppose that the DDH assumption holds, $\mathcal{FEME}$ is secure, $\mathcal{MAC}$ is unforgeable, and H is a random oracle, then PriSrv+ is a secure service discovery protocol and satisfies bilateral anonymity.

PriSrv+ vs PriSrv: Comparison

Protocol	Expressiveness			Security and Privacy				Usability		
	Monotonic Policy	Arbitrary Attribute	Large Universe	Privacy Broadcast	Mutual Authn.	Bilateral Anon.	Pub. Attri. Hidden	No Pre-reg. Pairing	No 3rd-party Dependence	No In-advance ID Issuance
PriSrv[14]	√	×	×	√	√	√	×	√	√	×
PriSrv+	√	√	√	√	√	√	√	√	√	√

Table 2: Comparison of Private Service Discovery Protocols with Bilateral Policy Control

Expressiveness

- PriSrv+: LSSS policies + large-universe, arbitrary strings
- PriSrv: small-universe + binary vectors (higher overhead as attributes grow)

Security & privacy

- Both protect discovery + mutual auth
- PriSrv leaks some public attributes/policies in outer layer
- PriSrv+ reveals only attribute names, hides all values

Usability

- Both avoid pre-registered pairing and third-party online support
- PriSrv requires credential issuance
- PriSrv+ removes credential lifecycle ⇒ simpler deployment

[14] 1, "PriSrv: Privacy-Enhanced and Highly Usable Service Discovery in Wireless Communications", 2024.

SMU SINGAPORE MANAGEMENT UNIVERSITY

FEME vs. ACME: Key Results (Big Picture)

Across all curves/security levels

- FEME has **substantially lower computation and ciphertext size** than ACME.
- As security increases (80→100-bit), both slow down, but **FEME's advantage persists**.

Headline numbers (averaged across curves)

- Setup: **1.74**× **faster**, |mpk| ↓ **71.68%**, |msk| ↓ **94.66%**.
- EKGen: **3.55**× **faster** (but |EK| is **larger** than ACME).
- DKGen: **3.39**× **faster**, |DK| ↓ **68.97%**.
- PolGen: **94.19**× **faster**, |SK| ↓ **98.07%**.
- Enc: **7.62**× **faster**.
- Dec: **6.23**× **faster**.
- |CT|: ↓ **87.33%** (average).

Summary FEME delivers **order-of-magnitude gains** in policy/key generation and **multi-**× **gains** in online Enc/Dec, while cutting ciphertext size by ≈ 87% (except a modest |EK| increase).

Curves	MNT159 (80-bit Security)		MNT201 (90-bit Security)		BN256 (100-bit Security)	
Schemes	ACME	FEME	ACME	FEME	ACME	FEME
Algorithms	Computation Costs (ms)					
Setup	20.526	8.411	26.882	9.699	33.344	11.402
EKGen	35.451	8.124	41.365	9.393	48.485	10.031
DKGen	21.630	4.150	18.640	3.836	15.750	4.787
PolGen	359.807	2.998	327.796	3.026	237.675	3.697
Enc	146.931	19.660	167.337	20.302	187.822	18.275
Dec	123.772	20.109	188.346	28.832	231.214	27.283
Algo.	Param.	Communication Costs (KB)					
Setup	|mpk|	1.044	0.344	1.332	0.428	4.128	1.071
Setup	|msk|	1.2	0.065	1.36	0.074	1.6	0.083
EKGen	|EK|	0.172	0.320	0.220	0.417	0.544	1.184
DKGen	|DK|	0.86	0.235	1.1	0.306	2.72	0.912
PolGen	|SK|	13.932	0.127	17.82	0.165	44.064	1.169
Enc	|CT|	164.34	23.287	212.964	29.599	537.984	63.104

Table 1: Performance of FEME and ACME (on Desktop)

SMU SINGAPORE MANAGEMENT UNIVERSITY

PriSrv+ vs. PriSrv: Setup, Platforms, and Summary Results

Same parameters. We use the same parameters as the FEME-vs-ACME table across the same three curves (MNT159 / MNT201 / BN256).

Four platforms (covering real SD environments).

- **Desktop:** Intel Core i9-7920X.
- **Laptop:** Intel Core i5-10210U.
- **Phone:** ARM Cortex (@ 2.4GHz).
- **Raspberry Pi:** ARM Cortex (@ 1.5GHz).

Two protocol phases measured (metrics: computation in ms, communication in KB).

- Privacy-preserving broadcast
- Anonymous mutual authentication

Computation improvements (average across platforms).

- **Broadcast phase:** PriSrv+ 54.336 ms vs. PriSrv 443.868 ms $\Rightarrow$ **7.17$\times$ faster**.
- **Mutual authentication:** PriSrv+ 233.181 ms vs. PriSrv 1008.02 ms $\Rightarrow$ **3.32$\times$ faster**.
- **End-to-end:** PriSrv+ 287.517 ms vs. PriSrv 1451.88 ms $\Rightarrow$ **4.05$\times$ faster**.

Communication reductions (dominated by ciphertext size).

- **Broadcast comm:** $\downarrow$ **87.33%**.
- **Auth comm:** $\downarrow$ **86.64%**.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Real Wireless Prototype: Scaling (Wi-Fi 802.1X, BN256)

Real-world integration

- Implemented in a Wi-Fi authentication environment (802.1X style).
- Client: ***wpa_supplicant***, Provider/AP side: ***hostapd***.
- Two laptops, curve: **BN256**.

Scaling experiment

- Attributes: $\{25, 50\}$.
- Policy share number: $\{3, 7, 11, 13, 17, 21, 27\}$ (proxy for access structure complexity).

Computation results (averages)

- PriSrv+: $T_B = 0.252\,\text{s},\ T_S = 0.166\,\text{s},\ T_C = 0.592\,\text{s}$.
- PriSrv: $T_B = 2.478\,\text{s},\ T_S = 0.374\,\text{s},\ T_C = 2.853\,\text{s}$.
- Speedups: **8.83× (broadcast)**, **1.25× (server)**, **3.82× (client)**.

Communication results (same scaling parameters)

- Server communication is constant and identical: $|\text{Server}| = 0.82\,\text{KB}$.
- PriSrv+: $|\text{Broadcast}| = 79.623\,\text{KB},\ |\text{Client}| = 83.64\,\text{KB}$.
- PriSrv: $|\text{Broadcast}| = 677.488\,\text{KB},\ |\text{Client}| = 681.505\,\text{KB}$.
- Reductions: **88.25% (broadcast)**, **87.73% (client)**.

SMU SINGAPORE MANAGEMENT UNIVERSITY

Conclusion

- PriSrv+ significantly advances privacy-preserving service discovery in wireless networks by introducing **Fast and Expressive Matchmaking Encryption (FEME)**.
- It overcomes the limitations of prior schemes by enabling **expressive bilateral access control** while enhancing **efficiency, security, privacy, and usability**.
- Evaluations demonstrate **notable gains in performance** and **reduced communication overhead**, making it well-suited for **resource-constrained devices**.
- With **formal security guarantees** and **compatibility with existing wireless protocols**, PriSrv+ effectively meets the privacy and security demands of wireless service discovery environments.

SMU SINGAPORE MANAGEMENT UNIVERSITY