ACT Update

Draft progress and a path to general state machines

Samuel Schlesinger

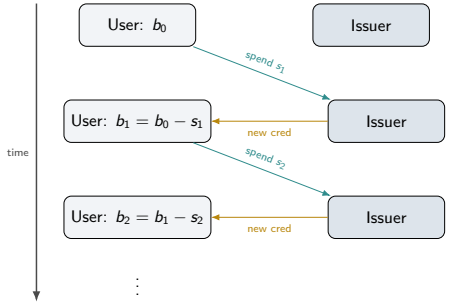
Privacy Pass Interim Meeting | May 13, 2026

What ACT is

A two-party protocol between a user and an issuer:

- User holds an *anonymous credential* carrying a hidden balance b.
- User presents an unlinkable spend proof that $b \geq s$; the issuer returns a refreshed credential.

The chain of refreshed credentials *is* the state machine.



What's changed in ACT since last time

Three updates, all landing in `draft-act` by IETF 126:

1. **Sigma proofs migration:** moving off bespoke sigma proofs onto the CFRG sigma proofs draft.
2. **SHAKE128** for hash-to-scalar and Fiat–Shamir, aligning with that draft.
3. **Flexible refunds:** the server credits back any $r \in [0, S_{\max}]$.

From scalar spend to general state machines

ACT today: $b' = b - s, \quad s \in [0, S_{\max}], \; b' \geq 0.$

Generalize the transition to a single *affine* step in a state vector:

$$\mathbf{s}' = T\,\mathbf{s} + \mathbf{u}, \quad \mathbf{s}, \mathbf{s}' \in \mathbb{Z}_q^k, \quad T \in \mathbb{Z}_q^{k \times k}, \quad \mathbf{u} \in \mathbb{Z}_q^k.$$

A valid transition proves, in zero knowledge:

- **Inputs:** each component of $\mathbf{u}$ satisfies its declared role (user-chosen, server-chosen, or jointly-determined) with its own range or equality bound.
- **Consequence:** $\mathbf{s}$ satisfies a conjunction of range / equality predicates over linear combinations of its entries.

Why? Epochs; access levels that control rate limits or rate consumption; and more generally, richer authorization policies expressed inside one credential.

Simple example: tiered per-spend cap

State vector $\mathbf{s} = (b,\ \ell)$: balance b and an immutable access tier ℓ baked in at issuance. Policy constants α, β.

Single admitted transition, server-chosen spend s:

$$\underbrace{\begin{pmatrix} b' \\ \ell' \end{pmatrix}}_{\mathbf{s}'} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{T} \underbrace{\begin{pmatrix} b \\ \ell \end{pmatrix}}_{\mathbf{s}} + \underbrace{\begin{pmatrix} -s \\ 0 \end{pmatrix}}_{\mathbf{u}(s)}.$$

Predicates proved in ZK (each a range proof on a linear combination):

- $0 \le s$ (server input is well-formed)
- $s \le \alpha\,\ell + \beta$ (per-spend cap scales with the hidden tier)
- $b' \ge 0$, i.e. $b - s \ge 0$ (solvency)

Questions

`draft-act`: github.com/samuelSchlesinger/draft-act