Late binding with PQ privacy

Christopher Patton

interim-2026-privacypass-01

Privacy Pass & PQ

- Privacy Pass enjoys security against ***store-now-decrypt-later attacks*****:** After Q-day, an attacker will be able to ***forge*** a Privacy Pass token (blind RSA or VOPRF) but won't be able to ***link*** redemptions to previous issuances
    - Privacy guarantee is information theoretic. For blind RSA, user sends $m' = mr^e \bmod N$ to issuer, and m' perfectly hides m. No computational assumptions required.
    - Forgery resistance requires assuming some problem (like factoring an RSA modulus) is computationally infeasible. Not so after Q-day.
- Quantum computers are coming, perhaps as early 2029. ***What should Privacy Pass do?***
    - My (revised) view: We must develop PQ alternatives. (Blind signatures / ACT-style credentials are on the horizon.) In the meantime, anything new that we deploy needs to at least be PQ unlinkable.
        - Token forgery is unlikely to be a target of the first quantum attacks. Impersonating google.com is much more valuable.

ACT & PQ

- ACT in its current form is PQ unlinkable: issuer sees a ***commitment*** to the nullifier `k` revealed during presentation (for double spend protection), but the commitment (`kG+rH`) perfectly hides `k`.
    - A quantum attacker could forge a credential by cracking the issuer's BBS key. (There are other attacks, but this is the most valuable one.)

ARC & PQ

- Similar story as ACT, except for ***late context binding:***
  - Tag (for double spend protection) is computed as $t=(k+nonce)^{-1}H(ctx)$, where:
    - k is used across presentations
    - $nonce$, ctx are unique for this presentation
    - $H(\cdot)$ hashes to the curve
  - ***Quantum privacy attack:*** Solve discrete log of t relative to base point $H(ctx)$. Attacker can now correlate presentations from the same credential holder:
    - $(\log_{H(ctx)} t)^{-1} = k+nonce$ for some k, $nonce$. Search for a tag $t' = (k+nonce')^{-1}H(ctx)$ for some $0 \leq nonce' < N$
  - Cheaper attacks might be possible: security of DY-PRF relies on stronger-than-usual assumptions [Orr25].

[Orr25] Michele Orrù. "Revisiting Keyed-Verification Anonymous Credentials." CCS 2025.

Late binding via information theoretic techniques

Credit: Michael Rosenberg and Michele Orrù (https://github.com/rozbb/everlasting-arc). I believe others (Watson Ladd?) talked about similar ideas in and around the list.

Key idea: Replace DY-PRF $t=(k+nonce)^{-1}H(ctx)$ with a ***universal hash function (UHF):***

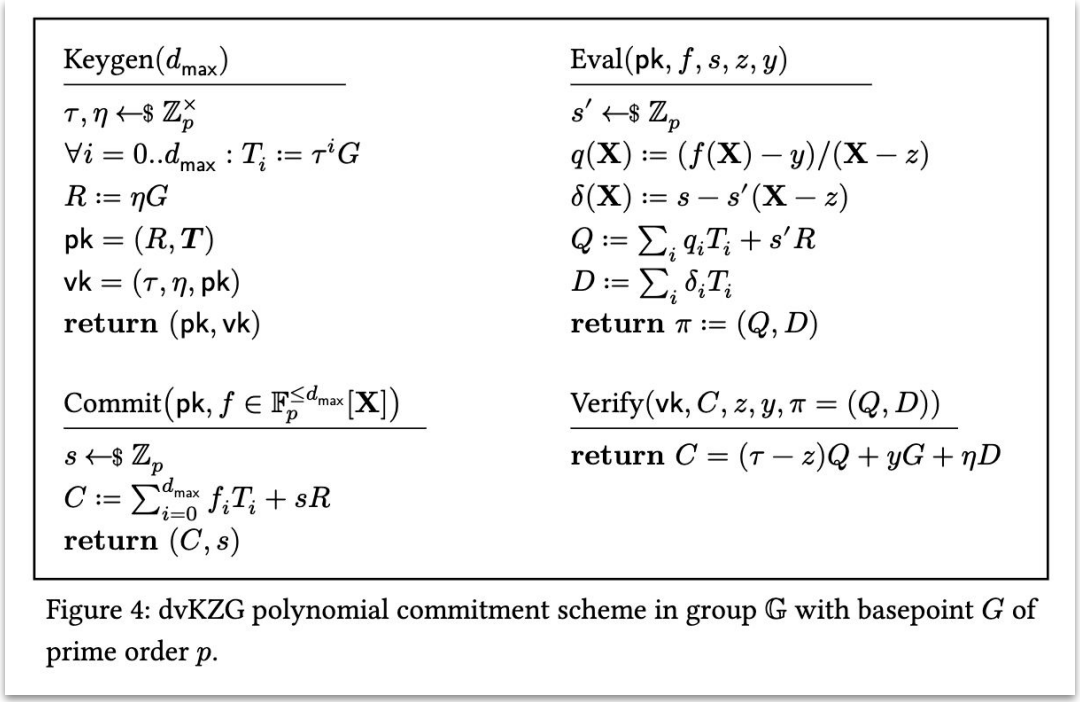
- During issuance, user commits to a degree-N polynomial f with random coefficients. During presentation, the tag is computed as $t=f(H(ctx)+nonce)$.
    - Perfectly hiding up to presentation limit N. (Caveat: Limit is the number of number of $nonce$, ctx pairs, not the number of $nonce$s per ctx).
- Using KZG [Orr25], communication cost is comparable to DY-PRF: $O(1)$, a group element for the commitment, plus a few additional constraints in Sigma proofs. But computation is ***high*** compared to DY-PRF: $O(N)$ rather than $O(\log N)$.

Caveats

- Due to computational cost, not useful for high presentation limit.
    - ACT with late binding might be a reasonable application: Use UHF to derive derive the first nullifier for a given presentation context; thereafter, manage the ACT state as usual.
- Write-up (https://github.com/rozbb/everlasting-arc) assumes BBS (a la ACT) is used instead of CMZ (a la ARc), presumably because BBS is more compatible with KZG, but I'm not sure.
- Michael seems confident on the design, but and end-to-end security analysis would be desirable. (For now we don't have concrete bounds for unforgeability or unlinkability.)
    - Incidentally: There are some differences in the security analyses of ARC and ACT that need to be reconciled in order to pick parameters.

designated verifier KZG

d_{max} is the presentation limit



$\underline{\text{Keygen}(d_{\mathsf{max}})}$

$\tau, \eta \leftarrow\$ \mathbb{Z}_p^\times$
$\forall i = 0..d_{\mathsf{max}} : T_i := \tau^i G$
$R := \eta G$
$\mathsf{pk} = (R, \boldsymbol{T})$
$\mathsf{vk} = (\tau, \eta, \mathsf{pk})$
return $(\mathsf{pk}, \mathsf{vk})$

$\underline{\text{Commit}(\mathsf{pk}, f \in \mathbb{F}_p^{\leq d_{\mathsf{max}}}[\mathbf{X}])}$

$s \leftarrow\$ \mathbb{Z}_p$
$C := \sum_{i=0}^{d_{\mathsf{max}}} f_i T_i + sR$
return (C, s)

$\underline{\text{Eval}(\mathsf{pk}, f, s, z, y)}$

$s' \leftarrow\$ \mathbb{Z}_p$
$q(\mathbf{X}) := (f(\mathbf{X}) - y)/(\mathbf{X} - z)$
$\delta(\mathbf{X}) := s - s'(\mathbf{X} - z)$
$Q := \sum_i q_i T_i + s'R$
$D := \sum_i \delta_i T_i$
return $\pi := (Q, D)$

$\underline{\text{Verify}(\mathsf{vk}, C, z, y, \pi = (Q, D))}$

return $C = (\tau - z)Q + yG + \eta D$

Figure 4: dvKZG polynomial commitment scheme in group $\mathbb{G}$ with basepoint G of prime order p.

Issuance

`ictx` stands for "issuance context" and is the same as ARC's `requestContext`

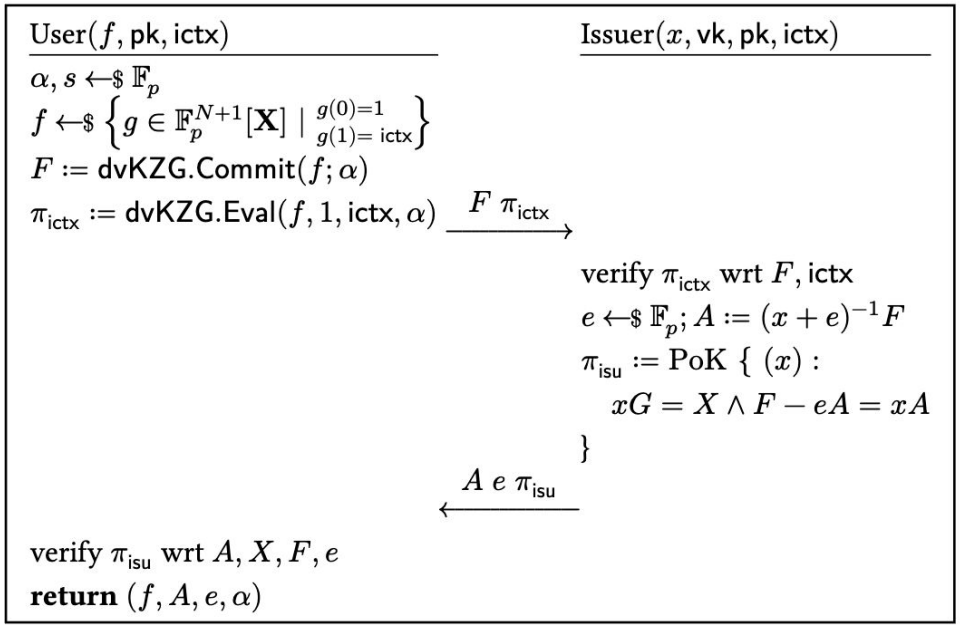


Figure 7: EL-ARC issuance

Presentation

`pctx` is the same as ARC's `presentationContext`

`ctr` is ARC's `nonce`

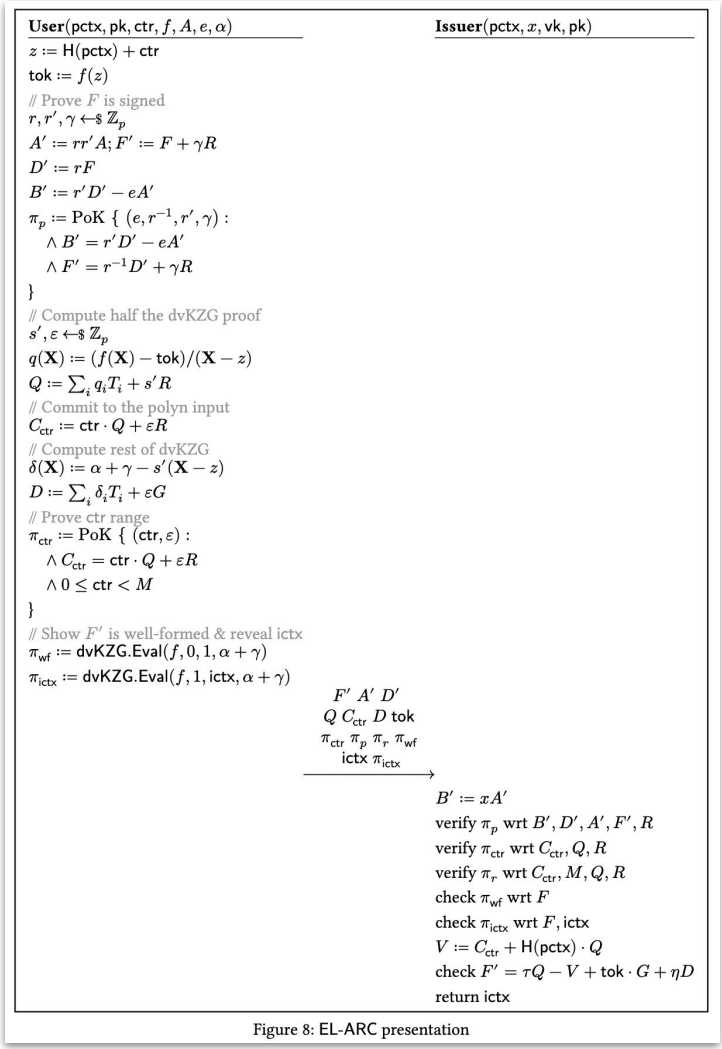


Figure 8: EL-ARC presentation