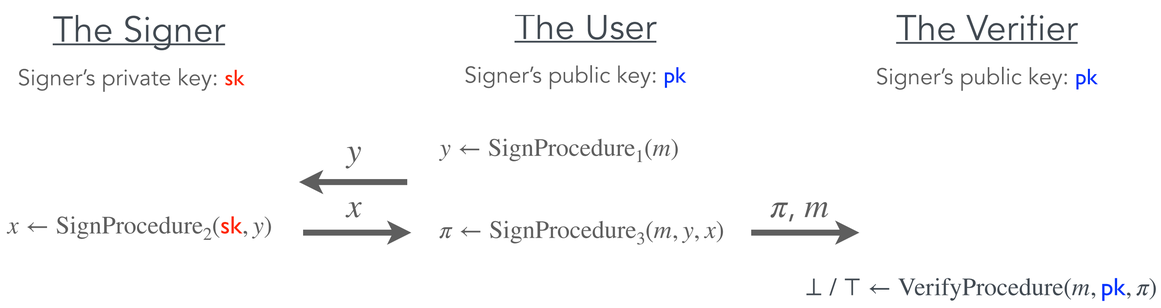
Blinding Post-Quantum Hash-and-Sign Signatures

Charles Bouillaguet, **Thibauld Feneuil**, Jules Maire,
Matthieu Rivain, Julia Sauvage, Damien Vergnaud

Sorbonne Université, Paris
CryptoExperts, Paris
Ecole Normale Supérieure-PSL, Paris

Blind signature scheme

Blind signature schemes define an interactive digital signature protocol between a user and a signer, ensuring that the signed message –and even the resulting signature– remain *unknown* to the signer; this property is referred to as *blindness*.



Round-optimal Blind Signature Protocol.

Fischlin's framework

The Fischlin framework [Fis06] offers a generic approach to constructing round-optimal blind signatures, utilizing standard signature schemes, commitment schemes and non-interactive zero-knowledge proofs (NIZKs).

$\text{SignProcedure}_1(m)$

- Sample randomness ρ
- $y \leftarrow \text{Commit}(m; \rho)$

$\text{SignProcedure}_2(\text{sk}, y)$

- $x \leftarrow \text{Sign}(\text{sk}, y)$

$\text{SignProcedure}_3(m, y, \rho, x, \text{pk})$

Using a NIZK, generate a proof transcript π proving knowledge of a digest y, commitment randomness ρ, and a signature x satisfying

$$\begin{cases} y = \text{Commit}(m; \rho) \\ \text{Verif}(\text{pk}, y, x) = \top \end{cases}$$

[Fis06] Fischlin. Round-optimal composable blind signatures in the common referee string model. CRYPTO 2006.

Hash-and-Sign signature schemes

The verification routine of post-quantum hash-and-sign signature schemes is largely algebraic. By design, there exists a value h' such that

$$h' = \text{Hash}(y) \quad \text{and} \quad \mathcal{G}(h') = x$$

where $\mathcal{G}$ is an algebraic function (e.g., a system of multivariate quadratic polynomials). The relation $\mathcal{G}(h') = x$ is typically very efficient to prove within a NIZK. It remains to prove that $h = \text{Hash}(h)$...

Removing the hash from the proved statement

[BLNS23] removes the need to prove the correctness of the hash computation within the statement, yielding a *lattice*-based blind signature scheme with signatures of about 22 KB.

$\text{SignProcedure}_1(m)$

- Sample small randomness r
- Compute $h = \text{Hash}(m, \text{Hash}(r))$
- $y \leftarrow Br + h$
- Using zk-SNARK, prove that y is well formed.

$\text{SignProcedure}_2(\text{sk}, y)$

- Verify the zk-SNARK proof.
- $x \leftarrow \mathcal{G}_{\text{sk}}^{-1}(y)$

$\text{SignProcedure}_3(m, y, \rho, x, \text{pk})$

- Using a NIZK, generate a proof transcript π' proving knowledge of (x, r) small such that

$$\mathcal{G}(x) = Br + h.$$

- $\pi := (\pi', h)$

Note: B is a public matrix.

[BLNS23] Beullens, Lyubashevsky, Nguyen, Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. CCS 2023.

Summary of our contributions

Base on the ideas of [BLNS23], we propose a new construction for blind signatures that offers several advantages over [BLNS23]:

- It enables the transformation of *any* post-quantum hash-and-sign signature scheme into a blind signature, independently of the underlying security assumptions (e.g., lattice-based, multivariate, etc.).
- The security of the resulting blind signature can be *reduced to that of the original signature scheme*, without introducing additional assumptions (beyond standard symmetric cryptographic primitives).
- It yields *efficient multivariate blind signatures* (5 – 8 KB) when applied to UOV-like signature schemes.

Our protocol

Rather than using a simple commitment scheme, our construction relies on a commit-append-and-prove scheme.

$\text{SignProcedure}_1(m)$

- Sample randomness r and ρ
- $c_r, \text{st} \leftarrow \text{CAP.Commit}(r; \rho)$
- Compute $h = \text{Hash}(m, c_r)$
- $y \leftarrow r + h$
- Using zk-SNARK, prove that y is well formed.

$\text{SignProcedure}_2(\text{sk}, y)$

- Verify the zk-SNARK proof.
- $x \leftarrow \mathcal{G}_{\text{sk}}^{-1}(y)$

$\text{SignProcedure}_3(m, y, \rho, x, \text{pk})$

- $c_x, \text{st} \leftarrow \text{CAP.Append}(x, \text{st})$
- Using the CAP proving routine, generate a proof transcript π' proving knowledge of x and r such that

$$\mathcal{G}(x) = r + h.$$

- $\pi := (c_r, c_x, \pi')$

$\text{VerifyProcedure}(m, \text{pk}, \pi)$

- Parse π as (c_r, c_x, π')
- Compute $h = \text{Hash}(m, c_r)$
- Check the transcript π'.

Remarks:

- A commit-append-and-prove scheme can be instantiated from a standard commit-and-prove scheme.
- Blindness follows from the fact that h is masked by r, together with the zero-knowledge properties of both the SNARK and the CAP NIZK.
- Proceeding as in [BLNS23] (i.e., without the commit-and-prove layer) enables the following forgery attack: an adversary can choose (m, μ) and find a pair (x, r) such that $\mathcal{G}(x) = r + h$, where $h = \text{Hash}(\mu, \rho)$. This attack is strictly easier than forging the underlying hash-and-sign signature.
- The main communication bottleneck between the signer and the user arises from the SNARK transcript; however, this data can be discarded immediately after it has been verified by the signer.

Application – multivariate blind signatures

Table: Comparison with more recent multivariate blind signature schemes.

	Same security as the original scheme	Signature Size	Communication Size
Our work	✔	5-8 KB	Few MB
[BFMR+25]	✘	7 KB	< 1 KB
[BBBMR26]	✔	24 KB	< 1 KB
[BBBMR26]	✘	6-7 KB	< 1 KB

[BFMR+25] Bouillaguet, Feneuil, Maire, Rivain, Sauvage, Vergnaud. Multivariate Commitment and Signatures with Efficient Protocols. ePrint 2025/2035.

[BBBMR26] Baum, Beckmann, Beullens, Mukherjee, Rechberger. Concretely Efficient Blind Signatures Based on VOLE-in-the-Head Proofs and the MAYO Trapdoor. ePrint 2026/109.