Improvements for the PERK Signature Scheme

PERK Team

Abstract

In this note, we share an ongoing work (to be made public soon) presenting some improvements for the PERK digital signature scheme. Similarly to the recent improvement of MIRA and RYDE [BFG+24], PERK can be improved by introducing a new MPCitH modeling for the PKP problem along with the VOLEitH proof system. Doing so, the PERK signature size is reduced and can reach 3.8 kB for NIST-1 security level. In addition, the proposed modification allows PERK to rely on on the standard PKP assumption rather than the currently used r-IPKP variant.

1 Introduction

The PERK signature scheme [ABB+23] is a Multi Party Computation in the Head (MPCitH) based scheme relying on the Permuted Kernel Problem (PKP). The PKP problem (along with the first PoK for PKP) was introduced by Shamir more than 30 years ago [Sha90] and has withstood cryptanalysis efforts since then [Geo92, BCCG93, PC94, JJ01, LP11, KMP19, PT21, SBC22]. PERK is the only candidate in the NIST's Post-Quantum Cryptography Standardization of Additional Digital Signature Schemes Project that relies on the PKP problem. As such, it provides some diversity in this new standardization effort which was one of the advertised goal of the NIST which is looking for "additional general-purpose signature schemes that are not based on structured lattices" [NIS22].

PERK is built from the combination of three components: (i) an MPCitH modeling for a PKP related assumption, (ii) an MPCitH based proof system and (iii) the Fiat-Shamir transform. In this note, we describe a new MPCitH modeling for the PKP problem based on a permutation matrix. Using this modeling along with the recent VOLEitH framework, the size of the PERK signatures can be significantly reduced as highlighted in Table 1. In addition, thanks to these modifications, the security of the scheme can be based on the standard PKP assumption (with the underlying field $\mathbb{F}_q$ being an extension of $\mathbb{F}_2$) while the current version relies on a relaxed PKP assumption denoted r-IPKP.

Overall, with this improvement, the PERK scheme has the following properties:

- **Security based on PKP.** PERK is the only candidate in the ongoing NIST's PQC Standardization effort relying on the PKP problem ;
- **Shorter signature sizes.** PERK has signatures ranging from 3.8 kB for its short instance to 4.6 kB for its fast instance for NIST-1 security level ;
- **Short key sizes.** PERK compressed keys are very short. In addition, the expanded public key of PERK is around 20 kB which might be an advantage in memory constrained environments.

Instance	**Security**	**Modeling**	**Proof System**	**Size (Sig.+pk)**
PERK (Original)	r-IPKP	Shared Permutation	MPCitH	**6.1 - 8.4 kB**
PERK-2 (New)	PKP	Permutation Matrix	VOLEitH	**3.9 - 4.7 kB**

Table 1: Expected modifications for PERK (sizes for NIST-1 security level)

2 The Permuted Kernel Problem

The security of PERK relies on the Permuted Kernel Problem (PKP). More precisely, the current version of PERK relies on a variant of PKP denoted r-IPKP for relaxed inhomegeneous PKP [ABB+23]. One should note that the proposed improvement for PERK relies on the standard PKP assumption (with $q = 2^k$) rather than the r-IPKP variant.

Definition 1 (Permuted Kernel Problem (PKP)) *Let (q, m, n) be positive integers such that $m < n$, $\boldsymbol{H} \in \mathbb{F}_q^{m\times n}$, $\boldsymbol{x} \in \mathbb{F}_q^n$ and $\pi \in \mathcal{S}_n$ be a permutation such that $\boldsymbol{H}\big(\pi[\boldsymbol{x}]\big) = \boldsymbol{0}$. Given $(\boldsymbol{H}, \boldsymbol{x})$, the Permuted Kernel Problem PKP(q, m, n) asks to find $\tilde{\pi} \in \mathcal{S}_n$ such that $\boldsymbol{H}\big(\tilde{\pi}[\boldsymbol{x}]\big) = \boldsymbol{0}$.*

Hereafter, we are interpreting the PKP problem in matrix form namely the secret permutation π is seen as a permutation matrix $\boldsymbol{P} \in \mathbb{F}_2^{n\times n}$ such that $\boldsymbol{HPx} = 0$

3 New MPCitH Modeling for PKP

Our new MPCitH modeling for PKP relies on permutation matrices. Informally, a permutation matrix is a square matrix of size n that features one 1 on each row as well as one 1 on each column and is populated with 0 on all the remaining coordinates. As such, it can be described by giving the n positions of the 1.

Given a secret position $i \in [0, n-1]$, our modeling build a vector over $\mathbb{F}_2$ of size n containing 1 in its ith coordinate and 0 in all the other coordinates. Doing this for each secret positions and arranging these vectors in a matrix, one get a matrix $\boldsymbol{P}$ over $\mathbb{F}_2$ containing one 1 on each row by construction. By checking that the sum of the coefficients of each column of $\boldsymbol{P}$ is equal to 1, one can verify that $\boldsymbol{P}$ is a permutation matrix.

Notations. Let n be a positive integer and let $d = \log_2(n)$. Let $\mathcal{B}^x : [0, n-1] \to \mathbb{F}_2^d$ be the function that given a value $x \in [0, n-1]$ returns the vector in $\mathbb{F}_2^d$ corresponding to its binary representation. Let $\overline{x} : \mathbb{F}_2^d \to \mathbb{F}_2^d$ be the function that given a vector $\boldsymbol{x} \in \mathbb{F}_2^d$ returns its complement namely $(1 \oplus x_0, \cdots, 1 \oplus x_{d-1})$.

Permutation Matrix Modeling for PKP. Let $\mathsf{pos} \in [0, n-1]^n$ be the positions of the 1 on each row of the secret permutation matrix. Given as input the vector $\boldsymbol{t} = (\mathcal{B}^{\mathsf{pos}_0}, \cdots, \mathcal{B}^{\mathsf{pos}_{n-1}}) \in (\mathbb{F}_2^d)^n$, compute $\boldsymbol{P} \in \mathbb{F}_2^{n \times n}$ as:

$$\forall (i, j) \in [0, n-1] \times [0, n-1],\ \boldsymbol{P}_{i,j} = \prod_{k=0}^{d-1} \left(\overline{\mathcal{B}_k^j} \oplus t_{i,k} \right).$$

One can verify that $\boldsymbol{P}$ is a solution of a given PKP instance $(\boldsymbol{H}, \boldsymbol{x})$ by checking that $\forall i \in [0, n-1], \sum_{j=0}^{n-1} \boldsymbol{P}_{i,j} = 1$, $\forall j \in [0, n-1], \sum_{i=0}^{n-1} \boldsymbol{P}_{i,j} = 1$ and $\boldsymbol{H}\boldsymbol{P}\boldsymbol{x} = \boldsymbol{0}$.

This new modeling presents two main benefits. Firstly, it allows to design a PoK for the PKP problem with input size $n \log_2(n)$ which represent a significant improvement with respect to the modeling currently used in PERK as illustrated in Table 2. Secondly, this modeling is compatible with linear and multiplicative sharing schemes contrarily to the modeling currently used in PERK. This allows to benefit from the recent improvements in MPCitH techniques as discussed in the next section.

Instance	**Modeling**	**Witness size**
PERK	Shared Permutation	$n \log_2(n) + n \log_2(q)$
PERK-2	Permutation Matrix	$n \log_2(n)$

Table 2: Witness size for different MPCitH modelings for the PKP problem.

4 The TCitH and VOLEitH Frameworks

The MPCitH paradigm [IKOS07] allows to build zero-knowledge proof systems using techniques from secure multi-party computation (MPC). It has been used extensively to design post-quantum signature schemes including PERK.

The MPCitH paradigm has been recently improved by the *VOLE-in-the-Head* (VOLEitH) [BBdSG+23b] and the *Threshold-Computation-in-the-Head* (TCitH) [FR23b, FR23a] frameworks. Using these frameworks, one can prove the knowledge of a witness satisfying some polynomial constraints. In addition, a recent work [BBM+24] has introduced a GGM tree related optimization that can be leveraged by both frameworks.

The VOLEitH and TCitH proof systems feature some similarities as explained in [FR23a] and lead to similar performances whenever the considered polynomial constraints are of degree 2 and the number of parties of the underlying MPC protocol is big enough. However, whenever the considered polynomial constraints feature a bigger degree, the VOLEitH framework is more interesting than the TCitH one. The MPCitH modeling for PKP proposed in Section 3 can be expressed easily as polynomial constraints of degree $\log_2(n)$. As such, the VOLEitH framework is more suited for this new modeling.

5 Resulting Improvements for PERK

5.1 Resulting sizes for PERK

The signature size of the PERK scheme can be significantly reduced using the modeling described in Section 3 along with the VOLEitH proof system discussed in Section 4 as illustrated in Table 3. Indeed, one can see that we are expecting to reduce the (public key + signature) size to the $[3.9 - 4.7]$ kB range.

Instance	N	Secret Key	Public Key	Signature
PERK (Fast)	32	16 B	0.3 kB	8.1 kB
PERK (Short)	256	16 B	0.3 kB	5.8 kB
PERK-2 (Fast)	256	16 B	0.1 kB	4.6 kB
PERK-2 (Short)	2048	16 B	0.1 kB	3.8 kB

Table 3: Expected improvement for PERK sizes for NIST security level 1.

5.2 Resulting performances for PERK

The MPCitH based schemes naturally feature a trade-off between signature size and performance. Although the impact of the proposed modification on performances is not known yet, we are expecting to reach better size/performance trade-offs than the current version of PERK as suggested by theoretical analysis [BFG+24], known VOLEitH implementations [BBdSG+23a, BBM+24] and the fact that the proposed modeling no longer require to sample and compose a large number of permutations. This explains why we consider values of N (number of parties involved in the emulated MPC protocol) as large as $N = 2048$ while the round 1 version of PERK only consider $N \leq 256$. We are currently working on an implementation demonstrating those effects and are expecting to reach better or similar performances than the current implementation of PERK.

6 Additional Applications of our Modeling

Our modeling relies on an efficient way to prove that a vector contains 1 in a given coordinate and 0 in all its remaining coordinates. While this is useful to prove that one knows a solution to a PKP instance as illustrated previously, this feature is also of independent interest. Indeed, it can be used to improve ring signatures [FR23a], PoK for the Regular Syndrome Decoding [CLY+24] and deniable authenticated KEM [GJK24] for instance. Additional applications of our modeling will be explored in the full version of this paper.

Acknowledgment

We would like to thank Thibauld Feneuil for pointing an error in a previous version of this note.

References

[ABB+23] Najwa Aaraj, Slim Bettaieb, Loïc Bidoux, Alessandro Budroni, Victor Dyseryn, Andre Esser, Philippe Gaborit, Mukul Kulkarni, Victor Mateu, Marco Palumbi, Lucas Perin, and Jean-Pierre Tillich. PERK. NIST's Post-Quantum Cryptography Standardization of Additional Digital Signature Schemes Project (Round 1), https://pqc-perk.org/, 2023.

[BBdSG+23a] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Sebastian Ramacher, Christian Rechberger, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. FAEST. NIST's Post-Quantum Cryptography Standardization of Additional Digital Signature Schemes Project (Round 1), https://faest.info/, 2023.

[BBdSG+23b] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 581–615, Cham, 2023. Springer Nature Switzerland.

[BBM$^+$24] Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. One tree to rule them all: Optimizing ggm trees and owfs for post-quantum signatures. Cryptology ePrint Archive, Report 2024/490, 2024. https://eprint.iacr.org/2024/490.

[BCCG93] Thierry Baritaud, Mireille Campana, Pascal Chauvaud, and Henri Gilbert. On the security of the permuted kernel identification scheme. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 305–311. Springer, Heidelberg, August 1993.

[BFG$^+$24] Loïc Bidoux, Thibauld Feneuil, Philippe Gaborit, Romaric Neveu, and Matthieu Rivain. Dual Support Decomposition in the Head: Shorter Signatures from Rank SD and MinRank. Cryptology ePrint Archive, Report 2024/541, 2024. https://eprint.iacr.org/2024/541.

[CLY$^+$24] Hongrui Cui, Hanlin Liu, Di Yan, Kang Yang, Yu Yu, and Kaiyi Zhang. Resolved: Shorter signatures from regular syndrome decoding and vole-in-the-head. In *IACR International Conference on Public-Key Cryptography*, pages 229–258. Springer, 2024.

[FR23a] Thibauld Feneuil and Matthieu Rivain. Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments. Cryptology ePrint Archive, Report 2023/1573, 2023. https://eprint.iacr.org/2023/1573.

[FR23b] Thibauld Feneuil and Matthieu Rivain. Threshold Linear Secret Sharing to the Rescue of MPC-in-the-Head. In *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*, 2023.

[Geo92] Jean Georgiades. Some remarks on the security of the identification scheme based on permuted kernels. *Journal of Cryptology*, 5(2):133–137, January 1992.

[GJK24] Phillip Gajland, Jonas Janneck, and Eike Kiltz. Ring Signatures for Deniable AKEM: Gandalf's Fellowship. *Cryptology ePrint Archive, Report 2024/890*, 2024.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 21–30, New York, NY, USA, 2007. Association for Computing Machinery.

[JJ01] Éliane Jaulmes and Antoine Joux. Cryptanalysis of PKP: A new approach. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 165–172. Springer, Heidelberg, February 2001.

[KMP19] Eliane Koussa, Gilles Macario-Rat, and Jacques Patarin. On the complexity of the permuted kernel problem. Cryptology ePrint Archive, Report 2019/412, 2019. https://eprint.iacr.org/2019/412.

[LP11] Rodolphe Lampe and Jacques Patarin. Analysis of some natural variants of the pkp algorithm. *Cryptology ePrint Archive*, 2011.

[NIS22] NIST. Request for additional digital signature schemes for the post-quantum cryptography standardization process, 2022. https://csrc.nist.gov/News/2022/request-additional-pqc-digital-signature-schemes.

[PC94] Jacques Patarin and Pascal Chauvaud. Improved algorithms for the permuted kernel problem. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 391–402. Springer, Heidelberg, August 1994.

[PT21] Thales Bandiera Paiva and Routo Terada. Cryptanalysis of the Binary Permuted Kernel Problem. In *International Conference on Applied Cryptography and Network Security*, pages 396–423. Springer, 2021.

[SBC22] Paolo Santini, Marco Baldi, and Franco Chiaraluce. Computational hardness of the permuted kernel and subcode equivalence problems. Cryptology ePrint Archive, Report 2022/1749, 2022. https://eprint.iacr.org/2022/1749.

[Sha90] Adi Shamir. An efficient identification scheme based on permuted kernels (extended abstract) (rump session). In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 606–609. Springer, Heidelberg, August 1990.