On Practical Post-Quantum Signatures from the Code Equivalence Problem

5th Pythagorean Conference

Edoardo Persichetti

2 June 2025

Department of Mathematical Sciences
Charles E. Schmidt College of Science
Florida Atlantic University

In This Talk

Roadmap

- Motivation and Background
- Bulding Signature Schemes
- Group Actions and (Code-Based) Cryptography
- Representation and Canonical Forms
- Conclusions

Roadmap

- ▶ Motivation and Background
- ▶ Bulding Signature Schemes
- ▶ Group Actions and (Code-Based) Cryptography
- ▶ Representation and Canonical Forms
- ▶ Conclusions

Post-Quantum Cryptography

In a few years time large-scale quantum computers might be reality. But then (Shor, '95):

- RSA
- DSA
- ECC
- Diffie-Hellman key exchange
- and many others ... **not secure** !

$\rightarrow$ NIST's Post-Quantum Cryptography Standardization Call (2017 - ongoing).

Main areas of research:

- Lattice-based cryptography.
- Hash-based cryptography.
- Code-based cryptography.
- Multivariate cryptography.
- Isogeny-based cryptography.

Traditional Code-Based Cryptography

Use hard problems from coding theory, such as the Syndrome Decoding Problem (SDP) in the Hamming metric.

For encryption, one can obtain a trapdoor by disguising the private code.

Example (McEliece/Niederreiter): use equivalent code.

$$G \longrightarrow SGP$$

The hardness of recovering the secret (S, P) depends on chosen code family.

This works well for encryption...
(Classic McEliece, BIKE, HQC)

...far less so for signature schemes.
(CFS, KKS, Stern,...)

History suggest that we have to do things a little differently.

Roadmap

- Motivation and Background
- Bulding Signature Schemes
- Group Actions and (Code-Based) Cryptography
- Representation and Canonical Forms
- Conclusions

Zero-Knowledge Protocols

An interactive protocol to prove knowledge of a secret...

...without revealing anything about it.

Prover (sk)		**Verifier (pk)**
Commit to some data	$\xrightarrow{cmt}$	
	$\xleftarrow{ch}$	Select random challenge
Produce response	$\xrightarrow{rsp}$	
	$(\ldots)$	Accept/Reject

- Completeness: honest prover always gets accepted.
- Soundness: dishonest prover (impersonator) has a bounded probability of succeeding.
- Zero-Knowledge: no information about the secret is leaked.

Example: Discrete Logarithms

Let g in a group G.

Witness is $sk = s$; instance given by $pk = g^s$.

Prover (sk)		**Verifier** (pk)
$cmt = g^r$	$\xrightarrow{cmt}$	
	$\xleftarrow{ch}$	$ch = c \xleftarrow{\$}$
$rsp = r - cs$	$\xrightarrow{rsp}$	
		$g^{rsp} \cdot pk^{ch} = cmt \xrightarrow{?}$ Accept/Reject

Verifying properties is obvious. Soundness depends on setting:
e.g. $c \in \{0, 1\}$ means error $1/2$.

Repetition is needed to bring soundness error down to desired target (e.g. 2^{-128}).

Can set $\sigma = (ch, rsp)$ and verify that $Hash(g^{rsp} \cdot pk^{ch}, msg) = ch$ (Schnorr).

ZKIDs can be turned into signature schemes using Fiat-Shamir transformation.

This method is very promising and usually leads to efficient schemes.
(Schnorr, 1989;...)

Strong security guarantees. No trapdoor is required!

For CBC, can avoid decoding: rely directly on SDP.

Use random codes and exploit hardness of finding low-weight words.
(Stern, 1993; ...)

High soundness error requires several repetitions to achieve security.

Due to protocol structure and nature of objects, this results in rather large signatures (e.g. > 20 kB for 128 sec. bits).

Idea: change the nature of the objects involved.

Roadmap

- Motivation and Background
- Bulding Signature Schemes
- Group Actions and (Code-Based) Cryptography
- Representation and Canonical Forms
- Conclusions

Group Action

Let X be a set and $(\mathsf{G}, \cdot)$ be a group. A group action is a mapping

$$\begin{array}{rccc} \star: & \mathsf{G} \times \mathsf{X} & \to & \mathsf{X} \\ & (g, x) & \mapsto & g \star x \end{array}$$

such that, for all $x \in \mathsf{X}$ and $g_1, g_2 \in \mathsf{G}$, $g_2 \star (g_1 \star x) = (g_2 \cdot g_1) \star x$.

The word cryptographic means that it has some properties of interest in cryptography, e.g.:

- Efficient evaluation, sampling and membership testing algorithms.
- A hard vectorization problem.

Group Action Vectorization Problem

Given the pair $x_1, x_2 \in \mathsf{X}$, find, if any, $g \in \mathsf{G}$ such that $g \star x_1 = x_2$.

Let X be a group of prime order p and G $= \mathbb{Z}_p^*$.

Then the vectorization problem is exactly DLP in X.

A huge amount of cryptography has been built using this simple, but very special group action!

Choosing the set X with this extra structure comes with several advantages and disadvantages.

- Useful properties (e.g. commutativity) and design options.
- Not post-quantum!

Recently, isogeny-based group actions have captivated the cryptographic scene, showing a unique performance profile.

What about group actions from coding theory?

Isometries in the Hamming Metric

Maps which preserve the distances (weights).

- Permutations: $\pi\big((a_1, a_2, \ldots, a_n)\big) = \big(a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)}\big)$.
- Monomials: permutations + scaling factors: $\mu = (v; \pi)$, with $v \in (\mathbb{F}_q^*)^n$

$$\mu\big((a_1, a_2, \ldots, a_n)\big) = \big(v_1 \cdot a_{\pi(1)}, v_2 \cdot a_{\pi(2)}, \ldots, v_n \cdot a_{\pi(n)}\big)$$

  Monomial matrix: permutation $\times$ diagonal.

- Monomials + field automorphism.

Two codes are equivalent if they are connected by an isometry.

We talk about permutation, linear and semilinear equivalence, respectively.

Code equivalence can be seen the action of a group G of isometries on linear codes.

Code-based Group Action

$$\begin{array}{rccc} \star: & \mathsf{G}\times\mathsf{X} & \to & \mathsf{X} \\ & (\psi,\mathscr{C}) & \mapsto & \psi(\mathscr{C}) \end{array}$$

where $\psi(\mathscr{C}) = \{\psi(\boldsymbol{c}) \mid \boldsymbol{c}\in\mathscr{C}\}$.

This view needs us to choose a standard representation for codes, e.g. systematic form.

In practice, we consider simply $RREF(GQ)$.

Then, code equivalence can be efficiently described using representatives, i.e. generator (or parity-check) matrices. Clearly:

$$\mathscr{C} \overset{\mathsf{PE}}{\sim} \mathscr{C}' \iff \exists\pi\in\mathsf{S}_n \text{ s.t. } G' = RREF(\pi(G)),$$
$$\mathscr{C} \overset{\mathsf{LE}}{\sim} \mathscr{C}' \iff \exists\mu\in\mathsf{M}_n \text{ s.t. } G' = RREF(\mu(G)).$$

where S_n is the symmetric group and $\mathsf{M}_n = \mathsf{M}_n(q)$ the monomial group.

The problem of deciding if two codes are equivalent is well-known in coding theory.

For our purpose, we are interested in the computational version: this is the vectorization problem for our action.

Permutation Equivalence Problem (PEP)

Given $\mathscr{C}, \mathscr{C}' \subseteq \mathbb{F}_q^n$, find a permutation π such that $\pi(\mathscr{C}) = \mathscr{C}'$.
In practice, given generators $G, G' \in \mathbb{F}_q^{k \times n}$, find $\pi \in \mathsf{S}_n$ such that

$$G' = RREF(\pi(G)).$$

Linear Equivalence Problem (LEP)

Given $\mathscr{C}, \mathscr{C}' \subseteq \mathbb{F}_q^n$, find a monomial μ such that $\mu(\mathscr{C}) = \mathscr{C}'$.
In practice, given generators $G, G' \in \mathbb{F}_q^{k \times n}$, find $\mu \in \mathsf{M}_n$ such that

$$G' = RREF(\mu(G)).$$

For practical applications, we are not interested in the semilinear version of the problem.

Could Code Equivalence be used as a stand-alone problem?

The situation for isometries recalls that of other group actions, such as for DLP (although without commutativity).

This means several existing constructions could be adapted to be based on Code Equivalence.

Possible to construct a ZK protocol based exclusively on the hardness of the code equivalence problem.
(Biasse, Micheli, P., Santini, 2020)

This can be then transformed into a full-fledged signature scheme via Fiat-Shamir.

Select hash function $Hash$.

Key Generation

- Choose random q-ary code $\mathscr{C}$, given by generator matrix G.
- sk: monomial map μ.
- pk: matrix $G' = RREF(\mu(G))$.

Prover		**Verifier**
Choose random monomial map $\tau \in \mathrm{M}_n$.		
Compute $\tilde{G} = RREF(\tau(G))$.		
Set $cmt = Hash(\tilde{G})$	$\xrightarrow{cmt}$	
	$\xleftarrow{b}$	Select random $b \in \{0, 1\}$.
If $b = 0$ set $rsp = \tau$	$\xrightarrow{rsp}$	Verify $Hash\big(RREF(rsp(G))\big) = cmt$.
If $b = 1$ set $rsp = \tau \circ \mu^{-1}$		Verify $Hash\big(RREF(rsp(G'))\big) = cmt$.

It is easy to prove completeness, 2-special soundness and honest-verifier zero-knowledge.

Before Fiat-Shamir, reduce soundness error $1/2 \implies t = \lambda$ parallel repetitions.

The protocol can be greatly improved with the following modifications:
(Barenghi, Biasse, P., Santini, 2021)

- Use multiple public keys and non-binary challenges.
  - \+ Lower soundness error: $1/2 \to 1/2^{\ell}$.
  - − Rapid increase in public key size.
- Use a challenge string with fixed weight w.
  - \+ Exploits imbalance in cost of response: seed vs monomial.
  - − Larger number of iterations.

Such modifications do not affect security, only requiring small tweaks in proofs or switching to equivalent security premises.

Security Considerations

PEP is not NP-complete, unless the polynomial hierarchy collapses.

(Petrank, Roth, 1997)

PEP is also deeply connected with Graph Isomorphism (GI) (reductions in both ways!), solvable in quasi-polynomial time.

At the same time, PEP is “not necessarily easy”.

(Petrank, Roth, 1997)

PEP is a special case of LEP; as a consequence, most solvers for PEP can be adapted to solve LEP as well, with different overhead depending on attack.

Efficient solvers for weak instances (e.g. small or trivial hull).

(Sendrier, 2000; Saeed-Taha, 2017; Bardet et al., 2020)

For general, hard instances, best solvers use combinatorial approach based on ISD.

(Leon, 1982; Beullens, 2020; Barenghi, Biasse, P., Santini, 2023)

Choose code parameters using latter type of attacks, following conservative criterion. Namely, pick n, k, q so that, for any d and any v, we have:

$$\sqrt{N_d(v)} \cdot C_{\text{ISD}}^{(d)}(n, k, q, v) > 2^\lambda.$$

For example for NIST Category 1 (≈ 128 sec. bits) we have $(n, k, q) = (252, 126, 127)$.

Protocol parameters (t, w, s) infer performance profile:

- $pk = (s-1)[k(n-k)\lceil \log_2(q) \rceil/8] + seed$ bytes
- $sig = w \cdot n\Big(\lceil \log_2(n) \rceil + \lceil \log_2(q-1) \rceil\Big)/8 + \{seeds, digest, salt\}$ bytes

Runtime is dominated by RREF computation, for both Keygen and Sign/Verify.

The protocol shows a high degree of flexibility, to cater for different priorities.

Roadmap

- Motivation and Background
- Bulding Signature Schemes
- Group Actions and (Code-Based) Cryptography
- Representation and Canonical Forms
- Conclusions

Equivalence Relations for Codes

We aim to provide an efficient representation for isometries.

Consider a subset $\mathsf{F} \subseteq \mathsf{G}$ of isometries and the equivalence relation induced by it.

This yields the equivalence classes $\mathfrak{C}_{\mathsf{F}}(\mathscr{C}) = \{\varphi(\mathscr{C}) \mid \varphi \in \mathsf{F}\}$.

If checking membership is efficient, then verifying that $\mathscr{C} \overset{\mathsf{LE}}{\sim} \mathscr{C}'$ can be done via any $\chi \in \mathsf{G}$ such that $\mathscr{C}^* = \chi(\mathscr{C}) \in \mathfrak{C}_{\mathsf{F}}(\mathscr{C}')$.

We can then look for a special choice for χ, one which allows a compact representation.

Indeed, if F is a subgroup, we can partition G into cosets, and we have

$$[\mathsf{G} : \mathsf{F}] = \frac{|\mathsf{G}|}{|\mathsf{F}|}.$$

This means the size of a witness is now

$$\log_2 [\mathsf{G} : \mathsf{F}] = \log_2 |\mathsf{G}| - \log_2 |\mathsf{F}|.$$

The goal is then to identify the largest F that fits the description.

Defining New Canonical Forms

Case 1: $F \simeq S_k \times S_{n-k}$. We use an ordering for multisets: sort rows, then columns.

This leads to:

$$[M_n : F] = \frac{|M_n|}{|F|} = \frac{n!(q-1)^n}{k!(n-k)!} = \binom{n}{k}(q-1)^n.$$

Case 2: $F \simeq M_k \times S_{n-k}$. We scale rows, then use Case 1 as subroutine to sort.

Here we have:

$$[M_n : F] = \frac{|M_n|}{|F|} = \frac{n!(q-1)^n}{k!(n-k)!(q-1)^k} = \binom{n}{k}(q-1)^{n-k}.$$

Case 3: $F \simeq M_k \times M_{n-k}$. We scale columns, then proceed as in Case 2.

Witness now is only:

$$[M_n : F] = \frac{|M_n|}{|F|} = \frac{n!(q-1)^n}{k!(n-k)!(q-1)^k(q-1)^{n-k}} = \binom{n}{k}.$$

Application to LESS

We provide bounds and verify that failure probability is negligible in all cases.

We modify the commitment step, where we commit to $Hash(\mathsf{CF}_{\mathsf{F}}(A))$.

A (carefully selected) coset representative can be used as rsp when $ch \neq 0$.

For LESS parameters, we have $\binom{n}{k} \leq n \cdot \mathcal{H}(R)$, where code rate $R = k/n = 1/2$
$\implies$ we can efficiently encode cosets with n bits.

As $n \approx 2\lambda$, this means signature size is now close to optimal!

The overhead for computing such canonical forms is very small compared to cost of RREF.
CF-LESS is shown to be still complete, 2-special sound and honest-verifier zero-knowledge.

We provide reductions between LEP and CF-LEP.

CF-LESS: Performance Overview

NIST Cat.	Code Params			Attack Factor	Prot. Params			pk (B)	sig (B)	CF
	n	k	q		s	t	w			
1	252	126	127	123.84	2	247	30	13939	8624 2481	- Case 3
					4	244	20	41785	5941 1846	- Case 3
3	400	200	127	197.67	2	759	33	35074	17208 5658	- Case 3
					4	244	20	105174	12768 4368	Case 3
5	548	274	127	271.56	2	1352	40	65792	30586 10036	- Case 3
					4	244	20	197312	25237 7769	- Case 3

Table: Impact of CF on LESS parameters. All sizes in bytes (B).

Roadmap

- Motivation and Background
- Bulding Signature Schemes
- Group Actions and (Code-Based) Cryptography
- Representation and Canonical Forms
- Conclusions

Conclusions

The introduction of the LESS scheme opened the way to an interesting application of isomorphism problems in cryptography.

The group action structure is fundamentally different from previous approach in code-based crypto.

Particularly suitable to develop protocols with advanced functionalities, e.g.:

- Ring signatures.
  (Barenghi, Biasse, Ngo, P., Santini, 2022)
- Threshold signatures.
  (Battagliola, Borin, Meneghetti, P., 2024)
- Blind signatures.
  (Kuchta, LeGrow, P., preprint)
- ...

Latest works drastically reduce signature size; smallest among code-based ZK schemes.

Still much work to do on performance (e.g. Gaussian elimination, pk size), functionalities (e.g. commutativity and other properties), applications etc.

Thank you for listening!
Any questions?

References

E. Berlekamp, R. McEliece, and H. Van Tilborg
On the inherent intractability of certain coding problems.
IEEE Transactions on Information Theory 24.3, 1978.

S. Barg
Some new NP-complete coding problems.
Problemy Peredachi Informatsii, 1994.

C.-P. Schnorr
Efficient identification and signatures for smart cards.
CRYPTO 1989.

J. Stern
A new identification scheme based on syndrome decoding.
CRYPTO 1993.

J.-F. Biasse, G. Micheli, E. Persichetti, and P. Santini
LESS is More: Code-Based Signatures Without Syndromes.
AFRICACRYPT 2020.

A. Barenghi, J.-F. Biasse, E. Persichetti, and P. Santini
LESS-FM: Fine-Tuning Signatures from the Code Equivalence Problem.
PQCRYPTO 2021.

E. Petrank and M. R. Roth
Is code equivalence easy to decide?
IEEE Transactions on Information Theory, 43(5):1602–1604, 1997.

References

N. Sendrier
The Support Splitting Algorithm.
IEEE Transactions on Information Theory, 1193–1203, 2000.

M. A. Saeed-Taha
Algebraic Approach for Code Equivalence.
PhD Thesis.

M. Bardet and A. Otmani and M. A. Saeed-Taha
Permutation Code Equivalence is Not Harder Than Graph Isomorphism When Hulls Are Trivial.
IEEE ISIT 2019.

J. Leon
Computing automorphism groups of error-correcting codes.
IEEE Transactions on Information Theory, 28(3):496–511, 1982.

W. Beullens
Not Enough LESS: An Improved Algorithm for Solving Code Equivalence Problems over $\mathbb{F}_q$.
SAC 2020.

A. Barenghi, J.-F. Biasse, E. Persichetti, and P. Santini
On the Computational Hardness of the Code Equivalence Problem in Cryptography.
Advances in Mathematics of Communications, 17(1):23–55, 2023.

References

E. Persichetti, and P. Santini
A New Formulation of the Linear Equivalence Problem and Shorter LESS Signatures.
ASIACRYPT 2023.

T. Chou, E. Persichetti, and P. Santini
On Linear Equivalence, Canonical Forms, and Digital Signatures.
Designs, Codes and Cryptography, 2025.

A. Barenghi, J.-F. Biasse, T. Ngo, E. Persichetti, and P. Santini
Advanced Signature Functionalities from the Code Equivalence Problem.
International Journal of Computer Mathematics: Computer Systems Theory, 2022.

M. Battagliola, G. Borin, A. Meneghetti and E. Persichetti
Cutting the GRASS: Threshold GRoup Action Signature Schemes.
CT-RSA 2024.

V. Kuchta, J. LeGrow, and E. Persichetti
Code-Based Blind Signatures.
preprint, to appear.