Developing Innovative Frameworks for Efficient Code-based Signatures

Edoardo Persichetti

14 July 2022

Charles E. Schmidt College of Science
Florida Atlantic University

In This Talk

- Introduction
- A Look into the Past
- New Frameworks
- Conclusions

Part I

Introduction

Post-Quantum Cryptography

In a few years time large-scale quantum computers might be reality.
But then (Shor, '94):

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

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

Main areas of research:

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

Motivation

Code-based cryptography has been doing really well for encryption/key establishment.

3 finalists in NIST's process:

- Classic McEliece (binary Goppa)
- BIKE (QC-MDPC)
- HQC (QC Random Codes)

The same cannot be said for code-based signatures.

Only 4 NIST submissions, all either broken or withdrawn.

Yet, signature schemes are a crucial component in cryptography.

Can we fix this?

Decoding Problems

In general, it is hard to decode random codes.

Problem (General Decoding)

Given: $G \in \mathbb{F}_q^{k \times n}$, $y \in \mathbb{F}_q^n$ and $w \in \mathbb{N}$.
Goal: find a word $e \in \mathbb{F}_q^n$ with $wt(e) \leq w$ such that $y - e = x \in \mathcal{C}_G$.

Easy to see this is equivalent to the following.

Problem (Syndrome Decoding)

Given: $H \in \mathbb{F}_q^{(n-k) \times n}$, $y \in \mathbb{F}_q^{(n-k)}$ and $w \in \mathbb{N}$.
Goal: find a word $e \in \mathbb{F}_q^n$ with $wt(e) \leq w$ such that $He^T = y$.

NP-Complete (Berlekamp, McEliece and Van Tilborg, 1978; Barg, 1994).

Unique solution when w is below a certain threshold.

Very well-studied, solid security understanding (ISD).

How to Do Code-Based Cryptography?

Choose a code family with efficient decoding algorithm associated to description Δ and hide the structure.

To get trapdoor, need one more ingredient.

Assumption (Code Indistinguishability)

It is possible to describe an error-correcting code via a matrix M which is indistinguishable from a randomly generated matrix of the same size.

Example: use change of basis $S \in GL(k, q)$ and permutation $P \in S_n$ to obtain equivalent code.

Hardness of assumption depends on chosen code family.

Part II

A Look into the Past

The Intuitive Idea

Use the traditional SDP-based trapdoor within hash-and-sign framework as in e.g. Full Domain Hash (RSA).

Given message *msg*, trapdoor OW function f and hash function **H**.

Create signature $\sigma = f^{-1}(\mathbf{H}(msg))$. Verify if $f(\sigma) = \mathbf{H}(msg)$.

For CBC, trapdoor is decoding: CFS scheme.
(Courtois, Finiasz, Sendrier, 2001)

...except, domain is not "full".

Complex sampling leads to slow signing, large keys and potential weaknesses.
(Bleichenbacher, 2009; Faugère Gauthier-Umana, Otmani, Perret, Tillich, 2013; Landais, Sendrier, 2012; Bernstein, Chou, Schwabe, 2013)

Recent renditions still exhibit very similar features.
(Debris-Alazard, Sendrier, Tillich, 2018)

Zero-Knowledge Identification Schemes

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

...without revealing anything about it.

Prover (SK)		Verifier (PK)
Commit to random data	$\xrightarrow{\mathit{com}}$	
	$\xleftarrow{\mathit{ch}}$	Select random challenge
Produce response	$\xrightarrow{\mathit{rsp}}$	
		Accept/Reject

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

Building Signatures

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

- Replace verifier's challenge with $\mathbf{H}(com, msg)$.
- Form signature as $\sigma = (com, rsp)$.
- Verify as in identification protocol.

This method for building signatures 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)

Stern's ZKID Protocol

Select hash function $\mathbf{H}$.

Key Generation

- Choose random binary code $\mathcal{C}$, given by parity-check matrix H.
- SK: $e \in \mathbb{F}_2^n$ of weight w.
- PK: the syndrome $s = He^T$.

Prover | Verifier

Choose $y \in \mathbb{F}_2^n$ and permutation π.
Set $c_1 = \mathbf{H}(\pi, Hy^T), c_2 = \mathbf{H}(\pi(y))$
$c_3 = \mathbf{H}(\pi(y+e))$

$\xrightarrow{c_1, c_2, c_3}$

$\xleftarrow{b}$ Select random $b \in \{0, 1, 2\}$.

If $b = 0$ set $rsp = (y, \pi)$ — Verify c_1, c_2.

If $b = 1$ set $rsp = (y + e, \pi)$ $\xrightarrow{rsp}$ — Verify c_1, c_3.

If $b = 2$ set $rsp = (\pi(y), \pi(e))$ — Verify c_2, c_3 and $wt(\pi(e)) = w$.

About Stern's ZKID

High soundness error implies that adversary has non-trivial cheating probability; for Stern's scheme, soundness error is 2/3.

This means several repetitions are necessary to amplify error and reach target authentication level.

Trasmitting the entire transcript produces a very long signature (e.g. $\geq$ 100 kB).

Several variants proposed over the years:

- Stern, 1993.
- Véron, 1996.
- Gaborit, Girault, 2007.
- Cayrel, Véron, El Yousfi, 2010.
- Aguilar, Gaborit, Schrek, 2011.
- ...

Goal: decreasing soundness error. For example, CVE scheme achieves $\frac{q}{2(q-1)} \approx 1/2$. Efficient for large finite fields.

Part III

New Frameworks

Decreasing the Soundness Error

Signature sizes still too large ($>$ 30 kB).

Decreasing the Soundness Error

Signature sizes still too large (> 30 kB).

The idea is still valid - need to use a different protocol.

DECREASING THE SOUNDNESS ERROR

Signature sizes still too large ($>$ 30 kB).

The idea is still valid - need to use a different protocol.

“MPC-in-the-head” approach used e.g. in Picnic.

(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Decreasing the Soundness Error

Signature sizes still too large ($>$ 30 kB).

The idea is still valid - need to use a different protocol.

"MPC-in-the-head" approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier's checks are independent from the secret.

Decreasing the Soundness Error

Signature sizes still too large ($>$ 30 kB).

The idea is still valid - need to use a different protocol.

"MPC-in-the-head" approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier's checks are independent from the secret.

These can be offloaded to a trusted setup ("helper").

Decreasing the Soundness Error

Signature sizes still too large (> 30 kB).

The idea is still valid - need to use a different protocol.

"MPC-in-the-head" approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier's checks are independent from the secret.

These can be offloaded to a trusted setup ("helper").

Preprocessing phase prepares auxiliary collection of samples (shares).

Decreasing the Soundness Error

Signature sizes still too large ($>$ 30 kB).

The idea is still valid - need to use a different protocol.

“MPC-in-the-head” approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier’s checks are independent from the secret.

These can be offloaded to a trusted setup (“helper”).

Preprocessing phase prepares auxiliary collection of samples (shares).

“Opening” of a subset does not compromise security...

DECREASING THE SOUNDNESS ERROR

Signature sizes still too large (> 30 kB).

The idea is still valid - need to use a different protocol.

"MPC-in-the-head" approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier's checks are independent from the secret.

These can be offloaded to a trusted setup ("helper").

Preprocessing phase prepares auxiliary collection of samples (shares).

"Opening" of a subset does not compromise security...

...but allows for much larger challenge space.

Decreasing the Soundness Error

Signature sizes still too large (> 30 kB).

The idea is still valid - need to use a different protocol.

"MPC-in-the-head" approach used e.g. in Picnic.
(Ishai, Kushilevitz, Ostrovsky, Sahai, 2007; Katz, Kolesnikov, Wang, 2018)

Some of the verifier's checks are independent from the secret.

These can be offloaded to a trusted setup ("helper").

Preprocessing phase prepares auxiliary collection of samples (shares).

"Opening" of a subset does not compromise security...

...but allows for much larger challenge space.

We can use this in CBC! For example, apply this to CVE setting.
(Gueron, P., Santini, 2020)

KeyGen: as in CVE, usual syndrome s, matrix H.

HELPER

- Generate random $y, \tilde{e} \in \mathbb{F}_q^n$, with $\tilde{e}$ of weight w, from seed.
- Compute $aux = \{\mathbf{Com}(y + c\tilde{e})\}_{c \in \mathbb{F}_q}$.
- Send seed to prover and aux to verifier.

PROVER — VERIFIER

Prover		Verifier
Regenerate $y, \tilde{e}$ from seed.		
Determine μ s.t. $e = \mu(\tilde{e})$		
$\alpha = \mathbf{Com}(\mu, H(\mu(y))^T)$	$\xrightarrow{\alpha}$	
	$\xleftarrow{c}$	Select random $c \in \mathbb{F}_q$.
$z = y + c\tilde{e}$	$\xrightarrow{z}$	
		Verify $\alpha = \mathbf{Com}(\mu, H(\mu(z))^T - cs)$.
		Verify $\mathbf{Com}(z)$ with corresponding value from aux.

Here the soundness error is $1/q$.

Producing a Signature Scheme

Use "cut-and-choose" technique to remove preprocessing.

Producing a Signature Scheme

Use "cut-and-choose" technique to remove preprocessing.

Executing 1 out of M setups produces soundness error $max\left(\frac{1}{M}, \frac{1}{q}\right)$.

Producing a Signature Scheme

Use “cut-and-choose” technique to remove preprocessing.

Executing 1 out of M setups produces soundness error $max\left(\frac{1}{M}, \frac{1}{q}\right)$.

Iterate as needed, then apply Fiat-Shamir.

Producing a Signature Scheme

Use "cut-and-choose" technique to remove preprocessing.

Executing 1 out of M setups produces soundness error $max\left(\frac{1}{M}, \frac{1}{q}\right)$.

Iterate as needed, then apply Fiat-Shamir.

Several optimizations are possible (e.g. Merkle trees, seed trees).

Producing a Signature Scheme

Use "cut-and-choose" technique to remove preprocessing.

Executing 1 out of M setups produces soundness error $max\left(\frac{1}{M}, \frac{1}{q}\right)$.

Iterate as needed, then apply Fiat-Shamir.

Several optimizations are possible (e.g. Merkle trees, seed trees).

Can potentially yield smaller signatures, at the cost of increased computation (signing/verification time).

Use “cut-and-choose” technique to remove preprocessing.

Executing 1 out of M setups produces soundness error $max\left(\frac{1}{M}, \frac{1}{q}\right)$.

Iterate as needed, then apply Fiat-Shamir.

Several optimizations are possible (e.g. Merkle trees, seed trees).

Can potentially yield smaller signatures, at the cost of increased computation (signing/verification time).

GPS scheme parameters ($\lambda = 128$, sizes in kB):

M	τ	q	n	k	w	PK	Sig
512	23	128	220	101	90	0.10	27.06
1024	19	256	207	93	90	0.11	23.98
2048	16	512	196	92	84	0.11	21.22
4096	14	1024	187	90	80	0.12	19.76

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$.
Then e_A uniquely determines e given s and H.

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$. Then e_A uniquely determines e given s and H.

Let $\mathbb{F}_q \subset \mathbb{F}_{\text{poly}}$ such that $n \leq |\mathbb{F}_{\text{poly}}|$ and take distinct $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{\text{poly}}$.

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$. Then e_A uniquely determines e given s and H.

Let $\mathbb{F}_q \subset \mathbb{F}_{\text{poly}}$ such that $n \leq |\mathbb{F}_{\text{poly}}|$ and take distinct $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{\text{poly}}$.

Build $S(X) \in \mathbb{F}_{\text{poly}}[X]$ via polynomial interpolation of the points (γ_i, e_i).

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$. Then e_A uniquely determines e given s and H.

Let $\mathbb{F}_q \subset \mathbb{F}_{\text{poly}}$ such that $n \leq |\mathbb{F}_{\text{poly}}|$ and take distinct $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{\text{poly}}$.

Build $S(X) \in \mathbb{F}_{\text{poly}}[X]$ via polynomial interpolation of the points (γ_i, e_i).

Build $Q(X) = \prod_{i \in E}(X - \gamma_i)$ where $E =${nonzero pos. of e}.

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$.
Then e_A uniquely determines e given s and H.

Let $\mathbb{F}_q \subset \mathbb{F}_{\text{poly}}$ such that $n \leq |\mathbb{F}_{\text{poly}}|$ and take distinct $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{\text{poly}}$.

Build $S(X) \in \mathbb{F}_{\text{poly}}[X]$ via polynomial interpolation of the points (γ_i, e_i).

Build $Q(X) = \prod_{i \in E}(X - \gamma_i)$ where $E = \{$nonzero pos. of $e\}$.

Then $deg(Q) = w$ and $wt(e) \leq w$ is equivalent to

$$Q \cdot S - P \cdot F = 0$$

where $F = \prod_{i=1}^{n}(X - \gamma_i)$ and $deg(P) \leq w - 1$.

Proving Hamming Weight via Polynomials

The GPS machinery is very expensive. Can we do better?

Observation: if $H = (H'|I_{n-k})$ write $e = (e_A, e_B)$, so $s = H(e_A, e_B)^T$.
Then e_A uniquely determines e given s and H.

Let $\mathbb{F}_q \subset \mathbb{F}_{\text{poly}}$ such that $n \leq |\mathbb{F}_{\text{poly}}|$ and take distinct $\gamma_1, \dots, \gamma_n \in \mathbb{F}_{\text{poly}}$.

Build $S(X) \in \mathbb{F}_{\text{poly}}[X]$ via polynomial interpolation of the points (γ_i, e_i).

Build $Q(X) = \prod_{i \in E}(X - \gamma_i)$ where $E = \{$nonzero pos. of $e\}$.

Then $deg(Q) = w$ and $wt(e) \leq w$ is equivalent to

$$Q \cdot S - P \cdot F = 0$$

where $F = \prod_{i=1}^{n}(X - \gamma_i)$ and $deg(P) \leq w - 1$.

This transforms SDP into a polynomial problem and completely avoids the need for an isometry.

(Feneuil, Joux, Rivain, 2022)

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Find $S^{(j)}(X)$ using points $(\gamma_i, e_i^{(j)})$, where $e^{(j)} = (e_1^{(j)}, \ldots, e_n^{(j)})$.

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Find $S^{(j)}(X)$ using points $(\gamma_i, e_i^{(j)})$, where $e^{(j)} = (e_1^{(j)}, \ldots, e_n^{(j)})$.

By the linearity of the Lagrange interpolation, $S(X) = \sum_{j=1}^{M} S^{(j)}(X)$.

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Find $S^{(j)}(X)$ using points $(\gamma_i, e_i^{(j)})$, where $e^{(j)} = (e_1^{(j)}, \ldots, e_n^{(j)})$.

By the linearity of the Lagrange interpolation, $S(X) = \sum_{j=1}^{M} S^{(j)}(X)$.

Write $Q(X) = \sum_{j=1}^{M} Q^{(j)}(X)$ and then $(P \cdot F)(X) = \sum_{j=1}^{M} (P \cdot F)^{(j)}(X)$.

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Find $S^{(j)}(X)$ using points $(\gamma_i, e_i^{(j)})$, where $e^{(j)} = (e_1^{(j)}, \dots, e_n^{(j)})$.

By the linearity of the Lagrange interpolation, $S(X) = \sum_{j=1}^{M} S^{(j)}(X)$.

Write $Q(X) = \sum_{j=1}^{M} Q^{(j)}(X)$ and then $(P \cdot F)(X) = \sum_{j=1}^{M} (P \cdot F)^{(j)}(X)$.

To verify that $Q(X)S(X) = (P \cdot F)(X)$, check $Q(r_l)S(r_l) = (P \cdot F)(r_l)$ for $1 \leq l \leq t$ and r_l elements of an extension field $\mathbb{F}_{\text{points}}$ of $\mathbb{F}_{\text{poly}}$ (Schwartz-Zippel lemma).

Proof of Knowledge (in a nutshell)

Create some shares $e_A = \sum_{j=1}^{M} e_A^{(j)}$ and hence $e = \sum_{j=1}^{M} e^{(j)}$.

Find $S^{(j)}(X)$ using points $(\gamma_i, e_i^{(j)})$, where $e^{(j)} = (e_1^{(j)}, \ldots, e_n^{(j)})$.

By the linearity of the Lagrange interpolation, $S(X) = \sum_{j=1}^{M} S^{(j)}(X)$.

Write $Q(X) = \sum_{j=1}^{M} Q^{(j)}(X)$ and then $(P \cdot F)(X) = \sum_{j=1}^{M} (P \cdot F)^{(j)}(X)$.

To verify that $Q(X)S(X) = (P \cdot F)(X)$, check $Q(r_l)S(r_l) = (P \cdot F)(r_l)$ for $1 \leq l \leq t$ and r_l elements of an extension field $\mathbb{F}_{\text{points}}$ of $\mathbb{F}_{\text{poly}}$ (Schwartz-Zippel lemma).

This is done directly on shares $Q^{(j)}(r_l)$, $S^{(j)}(r_l)$ and $(P \cdot F)^{(j)}(r_l)$, via standard MPC techniques to verify multiplication triple.

Considerations and Future Work

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Considerations and Future Work

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Performance is extremely competitive!

Considerations and Future Work

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Performance is extremely competitive!

Scheme parameters ($\lambda = 128$, sizes in kB):

M	τ	q	n	k	w	$\mathbb{F}_{\text{poly}}$	$\mathbb{F}_{\text{points}}$	PK	Sig
256	17	2	1280	640	132	2^{11}	2^{22}	0.96	11.2
256	17	2^8	256	128	80	2^8	2^{24}	0.15	8.5

Considerations and Future Work

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Performance is extremely competitive!

Scheme parameters ($\lambda = 128$, sizes in kB):

M	τ	q	n	k	w	$\mathbb{F}_{\text{poly}}$	$\mathbb{F}_{\text{points}}$	PK	Sig
256	17	2	1280	640	132	2^{11}	2^{22}	0.96	11.2
256	17	2^8	256	128	80	2^8	2^{24}	0.15	8.5

Possible improvement using linear complexity to avoid interpolation.

(P., Randrianarisoa, 2022)

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Performance is extremely competitive!

Scheme parameters ($\lambda = 128$, sizes in kB):

M	τ	q	n	k	w	$\mathbb{F}_{\text{poly}}$	$\mathbb{F}_{\text{points}}$	PK	Sig
256	17	2	1280	640	132	2^{11}	2^{22}	0.96	11.2
256	17	2^8	256	128	80	2^8	2^{24}	0.15	8.5

Possible improvement using linear complexity to avoid interpolation.
(P., Randrianarisoa, 2022)

q-ary parameters can be refined, leading to improved performance (e.g. Sig $\approx$ 7 kB).

Considerations and Future Work

Signature scheme obtained via usual means (cut-and-choose, repetition, Fiat-Shamir).

Performance is extremely competitive!

Scheme parameters ($\lambda = 128$, sizes in kB):

M	τ	q	n	k	w	$\mathbb{F}_{\text{poly}}$	$\mathbb{F}_{\text{points}}$	PK	Sig
256	17	2	1280	640	132	2^{11}	2^{22}	0.96	11.2
256	17	2^8	256	128	80	2^8	2^{24}	0.15	8.5

Possible improvement using linear complexity to avoid interpolation.
(P., Randrianarisoa, 2022)

q-ary parameters can be refined, leading to improved performance (e.g. Sig $\approx$ 7 kB).

Optimized implementation underway.

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

Code Equivalence

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

The group action defined by isometries on linear codes recalls the one used extensively in other areas of cryptography (e.g. DLP).

Code Equivalence

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

The group action defined by isometries on linear codes recalls the one used extensively in other areas of cryptography (e.g. DLP).

This means several existing constructions could be adapted to be based on code equivalence, with interesting results.

Code Equivalence

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

The group action defined by isometries on linear codes recalls the one used extensively in other areas of cryptography (e.g. DLP).

This means several existing constructions could be adapted to be based on code equivalence, with interesting results.

LESS stems from a ZK protocol based exclusively on the hardness of the Linear Equivalence Problem.

(Biasse, Micheli, P., Santini, 2020)

Code Equivalence

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

The group action defined by isometries on linear codes recalls the one used extensively in other areas of cryptography (e.g. DLP).

This means several existing constructions could be adapted to be based on code equivalence, with interesting results.

LESS stems from a ZK protocol based exclusively on the hardness of the Linear Equivalence Problem.
(Biasse, Micheli, P., Santini, 2020)

Protocol can be tweaked to increase efficiency (e.g. multiple public keys, fixed-weight challenges). (Barenghi, Biasse, P., Santini, 2021)

CODE EQUIVALENCE

The notion of code equivalence is implicit in McEliece; could it be used as a stand-alone problem?

The group action defined by isometries on linear codes recalls the one used extensively in other areas of cryptography (e.g. DLP).

This means several existing constructions could be adapted to be based on code equivalence, with interesting results.

LESS stems from a ZK protocol based exclusively on the hardness of the Linear Equivalence Problem.
(Biasse, Micheli, P., Santini, 2020)

Protocol can be tweaked to increase efficiency (e.g. multiple public keys, fixed-weight challenges). (Barenghi, Biasse, P., Santini, 2021)

Group action structure allows to achieve advanced functionalities (e.g. identity-based, ring signatures). (Barenghi, Biasse, Ngo, P., Santini, 2022)

LESS ZK Identification Scheme

Public data: hash function **H**, code $\mathcal{C}$ with generator G

Key Generation

- SK: invertible matrix S and monomial matrix Q.
- PK: matrix $G' = SGQ$ (can be systematic form).

Prover's Computation

- Choose random monomial matrix $\tilde{Q}$.
- Set $\tilde{G} = SystForm(G\tilde{Q})$ and $h = \mathbf{H}(\tilde{G})$.

  (After receiving challenge bit b).
- If $b = 0$ respond with $\tau = \tilde{Q}$.
- If $b = 1$ respond with $\tau = Q^{-1}\tilde{Q}$.

LESS ZK IDENTIFICATION SCHEME

Public data: hash function **H**, code $\mathcal{C}$ with generator G

KEY GENERATION

- SK: invertible matrix S and monomial matrix Q.
- PK: matrix $G' = SGQ$ (can be systematic form).

PROVER'S COMPUTATION

- Choose random monomial matrix $\tilde{Q}$.
- Set $\tilde{G} = SystForm(G\tilde{Q})$ and $h = \mathbf{H}(\tilde{G})$.

  (After receiving challenge bit b).
- If $b = 0$ respond with $\tau = \tilde{Q}$.
- If $b = 1$ respond with $\tau = Q^{-1}\tilde{Q}$.

LESS ZK Identification Scheme

Public data: hash function $\mathbf{H}$, code $\mathcal{C}$ with generator G

Key Generation

- SK: invertible matrix S and monomial matrix Q.
- PK: matrix $G' = SGQ$ (can be systematic form).

Prover's Computation

- Choose random monomial matrix $\tilde{Q}$.
- Set $\tilde{G} = SystForm(G\tilde{Q})$ and $h = \mathbf{H}(\tilde{G})$.

  (After receiving challenge bit b).
- If $b = 0$ respond with $\tau = \tilde{Q}$.
- If $b = 1$ respond with $\tau = Q^{-1}\tilde{Q}$.

LESS ZK Identification Scheme

Public data: hash function $\mathbf{H}$, code $\mathcal{C}$ with generator G

Key Generation

- SK: invertible matrix S and monomial matrix Q.
- PK: matrix $G' = SGQ$ (can be systematic form).

Prover's Computation

- Choose random monomial matrix $\tilde{Q}$.
- Set $\tilde{G} = SystForm(G\tilde{Q})$ and $h = \mathbf{H}(\tilde{G})$.

  (After receiving challenge bit b).
- If $b = 0$ respond with $\tau = \tilde{Q}$.
- If $b = 1$ respond with $\tau = Q^{-1}\tilde{Q}$.

Verifier's Computation

- If $b = 0$ verify that $\mathbf{H}(SystForm(G\tau)) = h$.
- If $b = 1$ verify that $\mathbf{H}(SystForm(G'\tau)) = h$.

Part IV

CONCLUSIONS

Intuition

Finally, viable solutions for code-based signatures begin to appear.

INTUITION

Finally, viable solutions for code-based signatures begin to appear.

NIST is not satisfied with current state-of-the art for signatures (only 2 finalists, both lattice-based).
$\rightarrow$ New call scheduled, deadline June 2023.

INTUITION

Finally, viable solutions for code-based signatures begin to appear.

NIST is not satisfied with current state-of-the art for signatures (only 2 finalists, both lattice-based).
$\rightarrow$ New call scheduled, deadline June 2023.

Design, optimize and implement previous approaches to prepare 2 new NIST submissions.

INTUITION

Finally, viable solutions for code-based signatures begin to appear.

NIST is not satisfied with current state-of-the art for signatures (only 2 finalists, both lattice-based).
$\rightarrow$ New call scheduled, deadline June 2023.

Design, optimize and implement previous approaches to prepare 2 new NIST submissions.

Study further advanced functionalities and applications (e.g. threshold signatures, multi-signatures).

Intuition

Finally, viable solutions for code-based signatures begin to appear.

NIST is not satisfied with current state-of-the art for signatures (only 2 finalists, both lattice-based).
$\rightarrow$ New call scheduled, deadline June 2023.

Design, optimize and implement previous approaches to prepare 2 new NIST submissions.

Study further advanced functionalities and applications (e.g. threshold signatures, multi-signatures).

Devise specialized implementations (e.g. hardware, microcontrollers, side-channel resistant).

Intuition

Finally, viable solutions for code-based signatures begin to appear.

NIST is not satisfied with current state-of-the art for signatures (only 2 finalists, both lattice-based).
$\rightarrow$ New call scheduled, deadline June 2023.

Design, optimize and implement previous approaches to prepare 2 new NIST submissions.

Study further advanced functionalities and applications (e.g. threshold signatures, multi-signatures).

Devise specialized implementations (e.g. hardware, microcontrollers, side-channel resistant).

Explore the connection between codes and other post-quantum areas; isometry-based crypto?

Grazie, Danke, Merci, Grazcha, Thank you
and Congratulations to Joachim!