UNIVERSITÀ DI TRENTO

LETSS sign together

Linear Equivalence Threshold Signature Scheme

Michele Battagliola, Giacomo Borin, Alessio Meneghetti

Università di Trento

23rd April 2023

Contents

Introduction

Linear Equivalence Problem

Problem (Linear Code Equivalence)

Let $\boldsymbol{G}, \boldsymbol{G'} \in \mathbb{F}_q^{k \times n}$ be the generator matrices for two $[n, k]_q$ codes. Determine whether the two codes are linearly equivalent or not, i.e. if there exists an invertible matrix $\boldsymbol{S} \in GL_k(q)$ and a monomial matrix $\boldsymbol{Q} \in M_n(q)$ such that $\boldsymbol{G}' = \boldsymbol{SGQ}$.

- Studied for over 40 years, with several instances still considered hard
- Unlikely to be NP-hard [9] but hard on the average-case
- A deep study of its hardness can be read in [3]

LESS signature scheme

Linear Equivalence Signature Scheme

Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, and Paolo Santini. *LESS-FM: Fine-tuning Signatures from the Code Equivalence Problem*. Cryptology ePrint Archive, Paper 2021/396. `https://eprint.iacr.org/2021/396`. 2021. URL: `https://eprint.iacr.org/2021/396`

- Based on the code equivalence problem
- Render the identification protocol via the Fiat Shamir transform
- Achieves interesting parameters

The Identification Protocol

$$
\begin{tikzcd}
C \arrow[r, "Q"] \arrow[dr, dashed, bend right, "Q\tilde{Q}"'] & C' \arrow[d, "\tilde{Q}"] \\
 & \tilde{C}
\end{tikzcd}
$$

Figure: Commutative diagram for the identification protocol

- Public parameters : $\mathfrak{C}, \mathfrak{C}'$
- Secret Key : $\boldsymbol{Q}$
- Commitment : $\tilde{\mathfrak{C}}$

The Identification Protocol

$$
\begin{tikzcd}
C \arrow[r, "Q"] \arrow[dr, dashed, bend right, "Q\tilde{Q}"] & C' \arrow[d, "\tilde{Q}"] \\
 & \tilde{C}
\end{tikzcd}
$$

Figure: Commutative diagram for the identification protocol

- On challenge 0 discloses $\tilde{\boldsymbol{Q}}$
- On challenge 1 discloses $\boldsymbol{Q}\tilde{\boldsymbol{Q}}$

A Generalized Version

Problem (Group Action Inverse Problem)

Let $(X, G, \cdot)$ be a group action. Given x and y in X, find, if there exists, an element $g \in G$ such that $x = g \cdot y$.

There exists other signature schemes based on group actions, some of them are CSIDH [7], Csi-fish [5], Calamari and Falafl [4].

Threshold signature schemes

A T out of N threshold signature scheme (TSS) is a scheme that split the secret key in a way that allows any subgroup of T out of N users to generate a signature, but this is infeasible for any smaller group.

A T out of N threshold signature scheme (TSS) is a scheme that split the secret key in a way that allows any subgroup of T out of N users to generate a signature, but this is infeasible for any smaller group.

Shamir Secret Sharing T out of N

To share a secret in a field $\mathbb{F}$ we need simply to consider a polynomial f of degree $T - 1$, then share to P_i the value $f(i)$. Through linear Lagrange interpolation T parties can recover the secret $f(0)$.

Threshold signature schemes

- the N out N case will be referred as *full-threshold*
- Nist call for MPTC [6]
- There exists, a threshold signature scheme for effective group action in [8]
- It requires to work in a cyclic group, true only for CSI-FiSh [5]
- The main problem is that in general and in particular for LESS the group isn't even abelian

Full-threshold scheme

Remark

To obtain a full-threshold scheme for LESS we need to modify the identification protocol in a way that N users can collaborate in order to prove the mutual knowledge of a secret key.

- The Verifier view of the prototocol should remain unchanged
- Via the Fiat-Sharmir transform we can obtain a threshold signature

Identification protocol diagram

$$
\begin{tikzcd}
C \arrow[r, "Q_N"] \arrow[rrrddd, dashed, bend right, "Q\tilde{Q} = Q_N \cdots Q_1 \cdot \tilde{Q}_1 \cdots \tilde{Q}_N"'] & C_N \arrow[r, dotted, no head] & C_2 \arrow[r, "Q_1"] & C_1 = C' \arrow[d, "\tilde{Q}_1"] \\
& & & \tilde{C}_1 \arrow[d, dotted, no head] \\
& & & \tilde{C}_{N-1} \arrow[d, "\tilde{Q}_N"] \\
& & & \tilde{C}_N = \tilde{C}
\end{tikzcd}
$$

Identification protocol

Public Data Parameters : $q, n, k \in \mathbb{N}$, matrix $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ and hash function H.		
Private Key : Monomial matrix $\boldsymbol{Q} = \boldsymbol{Q}_N \cdots \boldsymbol{Q}_1$ with $\boldsymbol{Q}_i \in M_n(q)$.		
Shares for P_i : Monomial matrix $\boldsymbol{Q}_i$		
Public Key : $\boldsymbol{G}' = \boldsymbol{SGQ}$.		
PROVERS		**VERIFIER**
Set $\tilde{\boldsymbol{G}} \leftarrow \boldsymbol{G}'$ and for $i = 1, ..., N$ do :		
P_i get $\tilde{\boldsymbol{Q}}_i \xleftarrow{\$} M_n(q)$ and set $\tilde{\boldsymbol{G}} \leftarrow \mathrm{SF}(\tilde{\boldsymbol{G}}\tilde{\boldsymbol{Q}}_i)$	$\xrightarrow{h}$	
Set $h = \mathrm{H}(\mathrm{SF}(\tilde{\boldsymbol{G}}))$.		
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}$.
If $b = 0$ then $\mu \leftarrow \tilde{\boldsymbol{Q}}$.		Accept if $\mathrm{H}(\mathrm{SF}(\boldsymbol{G}'\mu)) = h$.
If $b = 1$ then $\mu \leftarrow \boldsymbol{I}$.	$\xrightarrow{\mu}$	
for $i = 1, ..., N$ do :		
P_i set $\mu \leftarrow \boldsymbol{Q}_i \cdot \mu \cdot \tilde{\boldsymbol{Q}}_i$.		Accept if $\mathrm{H}(\mathrm{SF}(\boldsymbol{G}\mu)) = h$.

Figure: Full threshold identification protocol for the knowledge of the Private Key

General identification protocol

Public Data Parameters : Group $\mathcal{G}$ acting on $\mathcal{X}$ via $\circ$, element $X \in \mathcal{X}$ and hash function H.
Private Key : Group element $g = g_1 \cdots g_N$ with $g_i \in \mathcal{G}$.
Shares for P_i : Group element g_i
Public Key : $x' = g \circ x$.

PROVERS		**VERIFIER**
Set $\tilde{x} \leftarrow x'$ and for $i = 1, \dots, N$ do :		
P_i get $\tilde{g}_i \xleftarrow{\$} \mathcal{X}$ and set $\tilde{g} \leftarrow \tilde{g}_i \circ \tilde{x}$	$\xrightarrow{h}$	
Set $h = \mathrm{H}(\tilde{g})$.		
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}$.
If $b = 0$ then $\mu \leftarrow \tilde{g}$.		Accept if $\mathrm{H}(\mu \circ x') = h$.
If $b = 1$ then $\mu \leftarrow e$.	$\xrightarrow{\mu}$	
for $i = 1, \dots, N$ do :		
P_i set $\mu \leftarrow \tilde{g}_i \cdot \mu \cdot g_i$.		Accept if $\mathrm{H}(\mu \circ x) = h$.

Figure: Full threshold identification protocol for the knowledge of the Private Key

Public Key generation

Algorithm 1 `KeyGen`

Require: $q, n, k \in \mathbb{N}$, $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$.

Ensure: Public key $\boldsymbol{G}' = \mathrm{SF}(\boldsymbol{G}\boldsymbol{Q})$, each partecipant hold $\boldsymbol{Q}_i$ such that $\prod \boldsymbol{Q}_i = \boldsymbol{Q}$.

```
Each participant P_i chooses Q_i ∈ m_n(q) and S_i ∈ GL_k(q).
Set G_0 = G.
for i = 1 to N do
    P_i computes G_i = SF(G_{i-1} Q_i)
    P_i produces a ZKP proving the knowledge of Q_i
    P_i sends G_i to P_{i+1}
end for
return G' = G_N. The private key of P_i is Q_i.
```

Algorithm 2 `Sign`

Require: $q, n, k \in \mathbb{N}$, $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$, a security parameter λ, an hash function H with domain $\{0,1\}^\lambda$, a public key $(\boldsymbol{G}, \boldsymbol{G}' = \mathrm{SF}(\boldsymbol{GQ}))$ where SF stands for Systematic Form. The party P_i knows the (multiplicative) share $\boldsymbol{Q}_i$ of $\boldsymbol{Q} = \boldsymbol{Q}_1 \cdots \boldsymbol{Q}_N$.

Ensure: A valid LESS signature for the message m under the public key $(\boldsymbol{G}, \boldsymbol{G}')$.

for $j = 1$ to λ **do**
    Set $\boldsymbol{G}^j_{N+1} = \boldsymbol{G}'$
    for $i = N$ to 2 **do**
        P_i chooses $\tilde{\boldsymbol{Q}}^j_i \in M_n(q)$ and sends $\boldsymbol{G}^j_i = \mathrm{SF}(\boldsymbol{G}^j_{i+1}\tilde{\boldsymbol{Q}}^j_i)$ to P_{i-1}
    end for
    P_1 chooses $\tilde{\boldsymbol{Q}}^j_1 \in M_n(q)$ and sets $\boldsymbol{G}^j = \boldsymbol{G}^j_1 = \mathrm{SF}(\boldsymbol{G}^j_2\tilde{\boldsymbol{Q}}^j_1)$
end for
Compute $\mathtt{ch} = H(\boldsymbol{G}^1 || ... || \boldsymbol{G}^\lambda || \mathsf{m})$
for $j = 1$ to λ **do**
    if $\mathtt{ch}_j = 0$ **then** P_i discloses $\tilde{\boldsymbol{Q}}^j_i$ and $\mathtt{resp}_j = \tilde{\boldsymbol{Q}}^j_N \cdots \tilde{\boldsymbol{Q}}^j_1$ is then published
    else set $U_{N+1} = I$
        for $i = N$ to 2 **do**
            P_i computes $U_i = \boldsymbol{Q}_i U_{i+1} \tilde{\boldsymbol{Q}}^j_i$ and sends U_i to P_{i-1}
        end for
        P_1 computes $U_1 = \boldsymbol{Q}_1 U_2 \tilde{\boldsymbol{Q}}^j_1$ and publishes $\mathtt{resp}_j = U_1$
    end if
end for
$\mathtt{resp} = \mathtt{resp}_1 || ... || \mathtt{resp}_\lambda$

Proof of security

Security equivalence

Theorem

Under the hardness of the linear code equivalence problem and in the random oracle model, the LETSS signature scheme is existentially unforgeable under adaptive chosen-message attacks.

A scheme is said existentially unforgeable under adaptive chosen-message attacks if it is secure against an attacker which is allowed access to an arbitrary number of message/signature pairs of his choosing and tries to forge a signature for a non queried message

Sketch of the proof

Proof.

- We need to show that if an adversary $\mathbb{A}$ is able to forge the signature scheme controlling all but one player, then it is possible to build a simulator $\mathcal{S}$ that interacting with $\mathbb{A}$ is able to forge the centralized signature.
- We proved that we can simulate the procedure with $N = 2$ controlling only one of the users. The two strategies can then be merged to simulate the general case.

Sketch of the proof

Proof.

- For the simulation of the KeyGen we need to add a ZKP as in fig. 1 to stick the adversary to its secret value when controlling the user 2.
- For the simulation of the Sign algorithm we want to avoid using additional ZKP, thus we need to reprogram the random oracle. This technique, which is the basis for the proof of the Fiat Shamir Transform [1], allows the simulator to decide the challenge for the message ahead of time.

□

Public Data Parameters : $q, n, k \in \mathbb{N}$, matrices $\boldsymbol{G}_a, \boldsymbol{G}_b \in \mathbb{F}_q^{k\times n}$ and hash function H.
Private Key : Monomial matrix $\boldsymbol{Q} \in M_n(q)$.
Public Key : $\boldsymbol{G}'_a = \mathsf{SF}(\boldsymbol{G}_a\boldsymbol{Q})$ and $\boldsymbol{G}'_b = \mathsf{SF}(\boldsymbol{G}_b\boldsymbol{Q})$.

PROVER		**VERIFIER**
Choose $\tilde{\boldsymbol{Q}} \xleftarrow{\$} \mathbb{F}_q^{n\times n}$ and set:		
$\tilde{\boldsymbol{G}}_a = \boldsymbol{G}_a\tilde{\boldsymbol{Q}},\ \tilde{\boldsymbol{G}}_b = \boldsymbol{G}_b\tilde{\boldsymbol{Q}}.$	$\xrightarrow{h}$	
Set $h = \mathrm{H}(\mathrm{SF}(\tilde{\boldsymbol{G}}_a) \| \mathsf{SF}(\tilde{\boldsymbol{G}}_b))$.		
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}$.
If $b = 0$ then $\mu = \tilde{\boldsymbol{Q}}$.	$\xrightarrow{\mu}$	Accept if $\mathrm{H}(\mathsf{SF}(\boldsymbol{G}_a\mu) \| \mathsf{SF}(\boldsymbol{G}_b\mu)) = h$.
If $b = 1$ then $\mu = \boldsymbol{Q}^{-1}\tilde{\boldsymbol{Q}}$.		Accept if $\mathrm{H}(\mathsf{SF}(\boldsymbol{G}'_a\mu) \| \mathsf{SF}(\boldsymbol{G}'_b\mu)) = h$.

Figure: Identification protocol to prove that the Private Key is used for the calculation

General threshold scheme

Subset solution

Proposition

Given a pair (T, N) consider the integer $M = \binom{N}{T-1}$ and the family $\mathcal{I}$ containing all the M subsets of $\{1, ..., N\}$ of cardinality $N - T + 1$. After labeling $\mathcal{I}$ as $\{ I_1, ..., I_M\}$ and using as secret key $\boldsymbol{Q} = \boldsymbol{Q}_{I_1} \cdots \boldsymbol{Q}_{I_M}$ we can have a (T, N)-threshold signature scheme sending to each user P_i all the $\boldsymbol{Q}_I$ such that $I \ni i$.

- Easy solution, but the share sizes and the number of rounds are exponential in T.
- The security proof is a straightforward adaptation of that of the full-threshold case

Example of $(3, 4)$-scheme

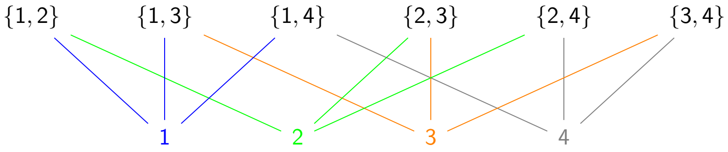

Figure: We have $6 = M = \binom{4}{2}$ subsets of cardinality $2 = N - T + 1$. Each user has $3 = \binom{3}{1} = \binom{N-1}{N-T}$ shares

Further directions and conclusions

Problems

- The lack of commutativity is a big obstacle to the generalization to the general case
- We want the share's sizes to be independent from T and N
- The group of monomial maps is not suitable for a secure secret sharing
- Secure multiparty computations solutions have been evaluated, but for now we are unable to exploit them in a meaningful way

Conclusions

- We have a full-threshold secure scheme, that generalise to other schemes based on group actions.
- Lack of commutativity poses a threat to the generalization.
- Combinatorics based solution is feasible only for small N.
- The use of abelian subgroups needs further investigations.

[1] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. "From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security". In: *Advances in Cryptology — EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 418–433. ISBN: 978-3-540-46035-0.

[2] Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, and Paolo Santini. *LESS-FM: Fine-tuning Signatures from the Code Equivalence Problem*. Cryptology ePrint Archive, Paper 2021/396. https://eprint.iacr.org/2021/396. 2021. URL: https://eprint.iacr.org/2021/396.

[3] Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, and Paolo Santini. *On the Computational Hardness of the Code Equivalence Problem in Cryptography*. Cryptology ePrint Archive, Paper 2022/967. https://eprint.iacr.org/2022/967. 2022. URL: https://eprint.iacr.org/2022/967.

[4] Ward Beullens, Shuichi Katsumata, and Federico Pintore. *Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices*. Cryptology ePrint Archive, Paper 2020/646. https://eprint.iacr.org/2020/646. 2020. URL: https://eprint.iacr.org/2020/646.

[5] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. *CSI-FiSh: Efficient Isogeny based Signatures through Class Group Computations*. Cryptology ePrint Archive, Paper 2019/498. https://eprint.iacr.org/2019/498. 2019. URL: https://eprint.iacr.org/2019/498.

[6] Luís T. A. N. Brandão, Michael Davidson, and Apostol Vassilev. *NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives*. Accessed: 2020-08-27. URL: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf.

[7] Wouter Castryck et al. *CSIDH: An Efficient Post-Quantum Commutative Group Action*. Cryptology ePrint Archive, Paper 2018/383. https://eprint.iacr.org/2018/383. 2018. URL: https://eprint.iacr.org/2018/383.

[8] Luca De Feo and Michael Meyer. *Threshold Schemes from Isogeny Assumptions*. Cryptology ePrint Archive, Paper 2019/1288. https://eprint.iacr.org/2019/1288. 2019. URL: https://eprint.iacr.org/2019/1288.

[9] E. Petrank and R.M. Roth. "Is code equivalence easy to decide?" In: *IEEE Transactions on Information Theory* 43.5 (1997), pp. 1602–1604. DOI: 10.1109/18.623157.

Table of contents