A code-based blind signature

Olivier Blazy, Philippe Gaborit, Julien Schrek, Nicolas Sendrier

A code-based blind signature

Olivier Blazy
Université de Limoges
olivier.blazy@unilim.fr

Philippe Gaborit
Université de Limoges
philippe.gaborit@xlim.fr

Julien Schrek
Université de Limoges
julien.schrek@xlim.fr

Nicolas Sendrier
INRIA
nicolas.sendrier@inria.fr

***Abstract*—In this paper we give the first blind signature protocol for code-based cryptography. Our approach is different from the classical original RSA based blind signature scheme, it is done in the spirit of the Fischlin approach [9] which is based on proofs of knowledge. To achieve our goal we consider a new tool for zero-knowledge (ZK) proofs, the Concatenated Stern ZK protocol, which permits to obtain an authentication protocol for concatenated matrices. A signature is then obtained from the usual Fiat-Shamir heuristic. We describe our blind signature protocol for cryptography based on Hamming metric and show how it can be extended to rank based cryptography. The security of our blind protocol is based on the security of a trapdoor function for the syndrome decoding problem: the CFS signature scheme for Hamming distance and on the more recent RankSign protocol for rank metric. We give proofs in the random oracle model (ROM) for our blind signature scheme, which rely on the Syndrome Decoding problem. The parameters we obtain for our protocol are practical for rank metric ($200kBytes$) for the signature length and $15kBytes$ for public key size) and a little less practical for Hamming distance.**

Keywords: Zero-knowledge protocols, coding theory, Stern SD scheme, CFS signature, code-based cryptography.

I. Introduction and motivation

Since the seminal work of Chaum on blind signature [4], this type of signature has found many applications in privacy preserving protocols like electronic voting protocol or electronic cash. The original blind signature scheme is based on the RSA signature protocol and many schemes have been proposed since, essentially based on number theory.

Our contribution: in this paper we propose the first generic blind signature protocol based on coding theory. The signature is based on a code-based trapdoor function which inverts a random syndrome.

Previous works Although there exist many blind signature scheme in classical cryptography based on number theory, there is only one published post-quantum blind signature protocol: the recent lattice based protocol of Ruckert [19], but this approach is based on the original RSA approach and cannot be not directly adaptated for Hamming or rank distance, because of the properties of the metric.

Organization of the paper The paper is organized as follows: Section 2 describes our security model for blind signatures, Section 3 recalls background on code-based cryptography, Section 4 describes a variation on Stern authentication protocol, Section 5 describes the protocol, Section 6 explains how the approach can be generalized to rank metric, and at last sections 6 and 7 study the security and the parameters of the protocol.

II. Security model for blind signature

As formalized in the paper by Pointcheval and Stern in [18], a blind signature scheme involves two parties, a user $\mathcal{U}$ and a Signer $\mathcal{S}$. The user submits a masked (or blinded) message that the Signer sign with a digital signature scheme whose public key is known. This part is named $\mathsf{BSProtocol}$. The user unmasks this signature to build a signature of the unmasked message which is valid for the Signer's public key. A verification can be made on the final signature with the Signer's public key.

More precisely, we can derive the definition from digital signatures. Instead of having a signing phase $\mathsf{Sign}(\mathsf{sk}, M; \mu)$ we have an interactive phase $\mathsf{BSProtocol}\langle\mathcal{S}, \mathcal{U}\rangle$ between the user $\mathcal{U}(\mathsf{vk}, M; \rho)$ who will (probably) transmit a masked information on M under some randomness ρ in order to obtain a signature valid under the verification key vk, and the signer $\mathcal{S}(\mathsf{sk}; \mu)$, who will generate something based on this value, and his secret key which should lead the user to a valid signature.

Such signatures are correct if when both the user and signer are honest then $\mathsf{BSProtocol}\langle\mathcal{S}, \mathcal{U}\rangle$ does indeed lead to valid signature on M under vk.

There are two additional security properties, one protecting the signer, the other the user.

- On one hand, there is an *Unforgeability* property, where a malicious user should not be able to compute $n + 1$ valid signatures on different messages after at most n interactions with the signer.
- On the other hand, the *Blindness* property says that a malicious signer who signed two messages M_0 and M_1 should not be able to decide which one was signed first.

$\mathsf{Exp}^{\mathsf{bl}-b}_{\mathcal{BS},\mathcal{S}^*}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{BSSetup}(1^\lambda)$
2. $(\mathsf{vk}, M_0, M_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{param})$
3. $\sigma_b \leftarrow \mathsf{BSProtocol}\langle\mathcal{A}, \mathcal{U}(\mathsf{vk}, M_b)\rangle$
4. $\sigma_{1-b} \leftarrow \mathsf{BSProtocol}\langle\mathcal{A}, \mathcal{U}(\mathsf{vk}, M_{1-b})\rangle$
5. $b^* \leftarrow \mathcal{S}^*(\texttt{GUESS} : M_0, M_1)$;
6. RETURN $b^* = b$.

$\mathsf{Exp}^{\mathsf{uf}}_{\mathcal{BS},\mathcal{U}^*}(\lambda)$
1. $(\mathsf{param}) \leftarrow \mathsf{BSSetup}(1^\lambda)$
2. $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{BSKeyGen}(\mathsf{param})$
3. For $i = 1, \ldots, q_s$, $\mathsf{BSProtocol}\langle\mathcal{S}(\mathsf{sk}), \mathcal{A}(\texttt{INIT} : \mathsf{vk})\rangle$
4. $\left((m_1, \sigma_1), \ldots, (m_{q_s+1}, \sigma_{q_s+1})\right) \leftarrow \mathcal{A}(\texttt{GUESS} : \mathsf{vk})$;
5. IF $\exists i \neq j, m_i = m_j$ OR $\exists i, \mathsf{Verif}(\mathsf{pk}, m_i, \sigma_i) = 0$
RETURN 0
6. ELSE RETURN 1

Fig. 1. Security Games for the *Blind Signatures*

III. Background on code-based cryptography

Let $\mathcal{F}$ denote the finite field with 2 elements and let $H \in \mathcal{F}^{(n-k)\times n}$ denote the parity-check matrix of some linear code of length n and dimension k. In this section and in all the paper $h()$ will denote a cryptographic hash function.

A. Syndrome Decoding

1) The CSD *Problem:* We consider the following problem:

Computational Syndrome Decoding Problem: Given a matrix $H \in \mathcal{F}^{(n-k)\times n}$, a word $u \in \mathcal{F}^{n-k}$, and an integer $w > 0$, find $x \in \mathcal{F}^n$ of Hamming weight $\leq w$ such that $H.x^T = u$.

This well known problem is NP-hard.

Decisional Syndrome Decoding Problem(D-SD): Given a matrix $H \in \mathcal{F}^{(n-k)\times n}$, an integer $w > 0$ a random word $x \in \mathcal{F}$ of weight w and a random syndrome s_2 of size $n-k$. Is it possible to distinguish between the random syndrome s_2 and the syndrome $s_1 = H.x^T$ associated to a small weight vector x ?

The previous DSD problem is proven as hard as its search version CSD in [3].

2) Gilbert-Varshamov Bound: The volume of a ball of radius w in the Hamming space $\mathcal{F}^n$ is $V_n(w) = \sum_{i=0}^{w}(q-1)^i\binom{n}{i}$. For given n and k, we call Gilbert-Varshamov (GV) bound is the integer b_{GV} such that $V_n(b_{GV}) \geq q^{n-k}$.

3) Complete Decoder: We call w*-bounded decoder* associated to H a procedure $\mathcal{F}^{n-k} \to \mathcal{F}^n$ which returns for all $u \in \mathcal{F}^{n-k}$ an element of $\mathrm{CSD}(H, u, w)$ (or fails if this set is empty). For given n and k and for almost all codes a w-bounded decoder fails for a proportion $\approx \exp(-V_n(w)/q^{n-k})$ of the instances. If we choose an integer $w > b_{GV}$, a w-bounded decoder almost never fails[1]. We will speak of a *complete*[2] decoder.

B. Trapdoor Digital Signature (CFS and Parallel-CFS)

Let w_0 be the smallest integer such that $\mathrm{CSD}(H, u, w_0) \neq \emptyset$ with high probability (i.e. from the previous section $w_0 = \lceil \tau_{\mathrm{gv}} \rceil$ or $\lceil \tau_{\mathrm{gv}} + 1 \rceil$). We assume here that the linear code defined by the parity check matrix H has some hidden algebraic structure (for instance a binary Goppa code) which enables a trapdoor complete w_0-bounded decoder D_H.

1) CFS.: Obtaining a practical complete decoder is not an easy task because the desired decoding bound w_0 is above the algebraic error correcting capability. It is possible for binary Goppa codes of high rate (*i.e.* the ratio k/n between dimension and length is close to 1) [5]: the resulting complete decoder is complex but still has an exponential advantage in complexity compared with the best generic algorithms for solving CSD.

> Let H be the parity check matrix of a CFS code, let D_H be the trap CFS decoding function The CFS problem is defined as given q access to a CFS oracle (Given x, it returns $y = D_H(x)$), and u^* the adversary has to return y^* such that $H(y^*)^\top = u^*$ and $wt(y^*) = w$ in polynomial time after at most q queries to the oracle, on words different from u^*.

Fig. 2. The CFS problem

[1]most of the time $w = b_{GV}$ is enough, exceptionally $w = b_{GV} + 1$

[2]the word *complete* is used here for convenience, the decoder may fail but for a negligible proportion of the instances

2) Security of CFS.: Parity check matrices of high rate Goppa codes can be distinguished from random matrices [6]. Still, this distinguishing attack does not lead to an efficient key recovery attack (recovering D_H from H), however it invalidates the security reduction given in [5]. We refer to [17] for more details on the security of CFS.

Limitations of CFS. The computational complexity of the CFS problem is super-polynomial in the size of the public key, so that the scheme leads to very high parameters, but still the scheme permits to construct usable difficult instances than can be used for cryptographic purposes.

3) Parallel-CFS.: It was proposed by Finiasz [8]. It consists in producing λ signatures (3 or 4) of related digests. If done correctly, the cost for an existential forgery attack can be made arbitrarily close to the cost for a universal forgery attack.

C. Stern's Authentication Protocols

1) Stern's protocol: Stern proposed in 1993 an authentication algorithm in [21]. In the protocol the prover $\mathcal{P}$ convinces the verifier $\mathcal{V}$ that he knows a secret word $s \in \mathcal{F}^n$ of weight w such that $u = H.s^T$ where $u \in \mathcal{F}^{n-k}$ and $H \in \mathcal{F}^{(n-k)\times n}$ are public. A fake prover has a probability $2/3$ to cheat at each iteration and thus many iterations are needed (137 for a cheating probability $< 2^{-80}$).

A *transcript* of the protocol for the pair (H, s), denoted $\mathrm{SP}(H, s)$, is a proof of knowledge that the prover knows a vector of weight w associated to the syndrome s for H, it consists in a transcript of the different steps of the protocol when the prover proves itself, like in the Fiat-Shamir heuristics through the use of random oracle which is used to simulate the challenges from the list of a sequence of commitments. Such a transcript is denoted by (H, u, w) and is said *valid* if it is a correct proof of knowledge of some $s \in \mathcal{F}^n$ of weight w such that $H.s^T = u$.

2) NIZK Digital Signature: The Fiat-Shamir NIZK (Non Interactive Zero Knowledge) paradigm [7] transform any ZK identification scheme into a digital signature. This digital signature essentially consist of a valid transcript of an execution of the protocol (see [20]).

IV. Concatenated Stern authentication protocol

In this section we introduce a (new) Concatenated Stern authentication protocol. This protocol is a simple randomized generalization of a non-randomized concatenated protocol given in [1]. We refer to [1] for a discussion on the notion of information leaking and testability of the Stern protocol for a non-randomized version of the protocol makes sense.

Let us consider Q a $k \times n_1$ binary matrix and R a $k \times n_2$ binary matrix. Suppose there exist a vector (x, y) with x and y of respective lengths n_1 and n_2 and of respective weight w_1 and w_2, and a syndrome s such that $[Q|R].(x, y)^T = s = Q.x^T + R.y^T$. The (randomized) concatenated Stern authentication protocol is a ZK protocol which permits to prove that the prover P knows a vector (x, y), for x and y of respective weight w_1 and w_2 such that $[Q|R].(x, y)^T = s = Q.x^T + R^T.y$. In the following S_n denotes the permutation

group of length n, and $|$ stands for concatenation. The protocol works as follows:

Concatenated Stern Zero-knowledge authentification protocol
Public data: two matrices: Q and R of respective size $k \times n_1$ and $k \times n_2$, a syndrome s.
Prover $\mathcal{P}$: a vector (x, y) for x and y of respective weight w_1 and w_2 such that $[Q|R].(x, y)^T = s = Q.x^T + R.y^T$.
The prover $\mathcal{P}$ interacts with a verifier $\mathcal{V}$ in 3 steps and a verification :

1) **Commitments** $\mathcal{P}$ generates : $\sigma_1 \xleftarrow{\$} S_{n_1}$, $\sigma_2 \xleftarrow{\$} S_{n_2}$, $u_1 \xleftarrow{\$} \mathcal{F}^n$,$u_2 \xleftarrow{\$} \mathcal{F}^{n'}$ and $r_1, r_2, r_3 \xleftarrow{\$} 1^\lambda$.
   $\mathcal{P}$ sends three commitments :
   $c_1 = h(\sigma_1(u_1)|\sigma_2(u_2)|r_1)$
   $c_2 = h(\sigma_1|Qu_1^T + Ru_2^T|\sigma_2|r_2)$
   $c_3 = h(\sigma_1(x + u_1)|\sigma(y + u_2)|r_3)$
2) **Challenge** $\mathcal{V}$ responds with $b \in \{0, 1, 2\}$.
3) **Answer** There are three possibilities:
   - If $b = 0$, P reveals $\sigma_1(u_1)$, $\sigma_2(u_2)$, $\sigma_1(x)$, $\sigma_2(y)$, r_1 and r_3.
   - If $b = 1$, he reveals σ_1, σ_2, $x + u_1$, $y + u_2$, r_2 and r_3.
   - If $b = 2$, he reveals σ_1, u_1, σ_2, u_2, r_1 and r_2.

   Verification
   - if $b = 0$ checks c_1 and c_3
   - if $b = 1$ checks c_2 and c_3
   - if $b = 2$ checks c_1 and c_2

Fig. 3. Concatenated Stern Zero-knowledge protocol

Theorem 1: The Concatenated Stern Zero-Knowledge protocol is a ZK protocol with 2/3 cheating probability.

Proof: The protocol we describe is an adaptation of the protocol described in [1] to which we added random values r_1, r_2 and r_3 so that the protocol cannot be testable and leaks non information (see [1] for details on testable Stern protocol). For verification the only non trivial check is (as for Stern original protocol) for $b = 1$ and the value c_1, which is checked with the public syndrome s, one recovers :$Qu_1^T + Ru_2^T$ as $Qu_1^T + Ru_2^T = Q(x + u_1)^T + R(y + u_2)^T - s$. The proof is straightforward from [1] with the ZK properties obtained from the random values r_1, r_2 and r_3. ■

V. A Code-Based Blind Signature protocol

A. High level overview:

The general idea of our protocol is inspired by an approach developped by Fischlin in [9], which works in three steps. In the first flow the user $\mathcal{U}$ chooses a random value x and consider Bx a committed value of x, he sends to the signer $\mathcal{S}$ a commitment $c = Commit(M, Bx)$ linking the message M and the random value x, then in a second flow $\mathcal{S}$ sends to $\mathcal{U}$, $y = TrapDoor(c)$, a preimage from its Trapdoor function, at last the last step, if a Zero-Knowledge proof of knowledge of the previous signature ($\Pi_{ZK}(x, y, Eval(y) = Commit(M, Bx))$), i.e. $\mathcal{U}$ knows the random value x and the value y obtained from $\mathcal{S}$, such that y is committed to x. In this way a Verifier is convinced that the user obtained a signature from the trapdoor function of $\mathcal{S}$ and this relatively to a hash involving a particular random value x, and the ZK proof permits to blind the identity of the user, who proves that he knows x and y without revealing them.

In term of coding theory, we define two random matrices A and B and a trapdoor matrix H. The first commitment step is obtained by choosing a small word x and sending a commitment $h(M|Bx^T) + Ax^T$ to $\mathcal{S}$, then $\mathcal{S}$ computes a preimage y by its trapdoor matrix H of the received commitment. The signature is then the couple: $B.x^T$ (a committed value of x) together with a ZK proof that $\mathcal{U}$ knows x and y such that :

$$\begin{pmatrix} A & H \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} h(M|B.x^t) \\ B.x^T \end{pmatrix},$$

in this way the Verifier is convinced that the user obtained a unique signature from $\mathcal{S}$ since on one hand y is committed to the random value x by $h(M|Bx^t)$ and since on the other hand x is chosen so that the syndrome $B.x^T$ fixes the chosen value x. Moreover the user's identity is blinded by the random value x and the general security of the scheme relies mainly on the secret trapdoor H.

B. Our protocol

The blind signature protocol is composed with three algorithms: a key generation algorithm, a blind signature algorithm, which involves the user $\mathcal{U}$, a message M and the signer $\mathcal{S}$ and a verification algorithm for the Verifier.

KeyGen(k, k′, n, n′) :
From some integer parameters k, k', n and n', he generates:
- H a trapdoor parity check matrix of size $k \times n$ and its trapdoor D_H, only available to the signer $\mathcal{S}$.
- A a random matrix of size $k \times n'$.
- B a random matrix of size $k' \times n'$.

Fig. 4. Key Generation

BSProtocol(w_x) :
1) **Blinding step**
The user $\mathcal{U}$:
- generates uniformly at random a vector x in $\mathcal{F}^{n'}$ of weight w_x.
- sends $\mu = h(M|Bx^T) + Ax^T$ to $\mathcal{S}$.

The signer $\mathcal{S}$ returns $y = D_H(\mu)$ of weight w_y to the user $\mathcal{U}$. If $D_H(\mu)$ returns $\perp$, then $\mathcal{U}$ can send an other request.

2) **Blind Signature step** : $\mathcal{U}$ sends the couple $(B.x^T, PoK)$
where PoK is a transcript of the proof of knowledge that $\mathcal{U}$ knows a vector (x, y) of weight w_x and w_y such that:

$$\begin{pmatrix} A & H \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} h(M|B.x^t) \\ B.x^T \end{pmatrix}$$

The proof of knowledge is obtained through the Concatenated ZK Stern protocol of Section 4, by taking $Q = \begin{pmatrix} \mathrm{A} \\ \mathrm{B} \end{pmatrix}$ and $\mathcal{R} = \begin{pmatrix} \mathrm{H} \\ 0 \end{pmatrix}$.

Fig. 5. The blind signature protocol

Verification: upon receiving the message M and and the signature $(B.x^T, Pok)$, the Verifier checks that the proof of knowledge PoK is correct and hence that the weights w_x and w_y of x and y are correct.

Fig. 6. Verification protocol

VI. Security

In this section we consider the security of our protocol, there are two properties to satisfy: the unforgeability property and the blindness property. We consider these two properties in the following theorems.

Theorem 2 (Unforgeability): If there exists an adversary against the soundness of the Blind Signature, then there exists

an adversary under either the CFS Problem, the Syndrome Decoding Problem, or the Soundness of the underlying Zero-Knowledge proof.

Proof: If an adversary $\mathcal{A}$ can win the game of unforgeability of the blind signature, then he can produce $N+1$ blind signatures with N requests to the blind oracle.

To exploit this adversary, we build a simulator in the following way. We first receive, the matrix H and a hash function h from the challenge oracle for CFS, and generate normally the other parameter of our blind signature.

Receiving signing queries, on string c_i, we forward it to the CFS oracle, and receive y_i such that $Hy_i^\top = c_i$.

Receiving hash queries, the simulator answers with a random value, and stores it to answer the same way to similar queries.

After at most q signing queries and n random oracle queries, the adversary sends us $q+1$ signature σ_j on messages m_j, by sending us values B_j and Zero-Knowledge proofs, that he knows x_j, y_j such that $B_j = Bx_j^\top$ and $Hy_j^\top = h(M_j|B_j) - A(x_j)^\top$.

As this is a valid forgery against the blind signature scheme, then all the $q+1$ signatures are valid.

Either the adversary manage to break the Soundness of one of the proof, or using the Random Oracle the Simulator manages to extract the values x_j, y_j.

If two values y_{j_1} and y_{j_2} are equal, then the adversary has managed to find a collision on $h(M_{j_b}|B_{j_b}) - A(x_{j_b})^\top$, in this case, we simply rewind to the furthest random oracle query on $M_{j_b}|B_{j_b}$ and output another random value such that there is no longer a collision (neither with the query corresponding to j_b, nor with any queries done before to the ROM). The forking lemma ensures us that the adversary advantages is approximately the same, after k rewinding where k is upperbounded by $\min(n, q)$.

After this, we are sure that all the $q+1$ values y_j are different, so there exists at least one y_j that does not come from the Challenge oracle. Rewinding one last time, and setting $h(M_{y_j}|B_j)$ to $u^* - A_{x_j}$, and invoking the forking lemmas, allows to recover an y_j such that $H(y_j)^\top = h(u^*)$ and so it allows to solve the CFS Challenge.

■

Theorem 3 (Blindness): If there exists an adversary against the Blindness of the Blind Signature, then there exists an adversary under the Zero-Knowledge property of the Stern protocol or the Computational Syndrome Decoding Problem.

Proof: If an adversary $\mathcal{A}$ can win the game of blindness of the blind signature, then he can break the decisional Syndrome Decoding problem.

To exploit this adversary, we build a simulator in the following way. We first receive a Decisional Syndrome decoding C, s and has to guess whether there exists a small x such that $C.x^\top = s$.

The simulator splits the matrix C into A and B, generates a matrix H honestly and publishes them as the public keys of the scheme, and gives H's trapdoor to the adversary.

The adversary then sends two messages M_0 and M_1 to the simulator. The simulator picks a random bit b, and proceeds to send the requests on M_b and M_{1-b}, and then outputs the signature on M_0.

With advantage ϵ, the adversary guesses whether $b = 0$ or not.

The simulator then proceeds to a sequence of games, first in the final signature, he proceeds in simulating the Zero-knowledge proof (still on an honest value) which leads to game G_1.

At this step, the adversary view is then B_{x_0}, and $h(M_b|B_{x_b}) + A_{x_b}$, so B_{x_0}, A_{x_0} and $A_{x_1} + h(M_0|B_{x_0}) + h(M_1|B_{x_1}) \neq A_{x_0}$ (Controlling the ROM allows to make sure of that, anyway it happens with overwhelming probability).

The simulator then splits s into s_1, s_2, sets $A_{x_0} = s_1, B_{x_0} = s_2$ If the answer to the challenge was yes, we are still in the previous game G_1. On the contrary, if it was no, it leads us to the last game G_2, where B_{x_0} is taken at random independently from A_{x_0}.

The last game G_2 only view a completely simulated answer, with random public values, so the adversary has no advantage against the blindness in G_2. The difference between G_2 and G_1 is the Decisional Syndrome Decoding problem, while the Zero-knowledge property differentiate G_1 from the real game. Hence $\epsilon \leq \mathsf{Adv}_{ZK} + \mathsf{Adv}_{DSD}$. Now the DSD problem is proven harder than the CSD problem in [3]. ■

VII. Blind signature for rank metric

In this section we outline how our blind signature protocol can be adapted for rank metric. Consider $x(x_1, ..., x_n)$ an element of $GF(q^m)^n$ and let $\mathcal{B}$ be a $GF(q)$-basis of $GF(q^m)$. Writing all the x_i in the basis $\mathcal{B}$, a $m \times n$ matrix X can be associated to x. The rank weight of x is then the rank of X. Rank-based cryptography is very similar to code-based cryptography although the first rank-based protocol appeared only in 1991, in both case a code structure is used, the only difference being the metric considered. In practice most of the concept and ideas developed for Hamming distance can be extended straighforwardly to rank metric. The problem of Syndrome Decoding(SD) becomes Rank Syndrome Decoding (RSD), which has been recently shown hard in [15] and also in its Decisional version [10]. We refer to [12] and the references within, for more details on rank-based cryptography. The main interest of rank-based cryptography is that the best known algorithms for solving the RSD problem have very high complexity(see [11]), so that in practice it permits to obtain very small size of keys (a few thousand bits) for hard to solve instances, when such sizes of keys can only be reached with additional structure (like cyclicity for instance) for code-based cryptography (Hamming distance) or lattice-based cryptography (Euclidean distance).

Most of cryptographic protocols have equivalent counterparts in rank metric and in particular all the tools used for our blind signature scheme have equivalent counterparts in rank metric : a rank based Stern authentication protocol has been developed in [14], and there is an equivalent notion of

concatenated rank Stern protocol in [2], at last there is also a trapdoor signature protocol: the RankSign protocol developed in [13], which has not a super-polynomial complexity at the difference of the CFS protocol.

Overall, even if we do not have space to describe it precisely in this short paper, our protocol and its proofs can be directly adapted for rank-based cryptography, simply replacing the code-based cryptographic algorithms by they rank-based counterparts since the general coding structure remains identical.

VIII. Instantiation and parameters

A. Instantiation

Overall the best practical attacks against forgery is an attack against the invertible trapdoor function D_H, CFS for Hamming distance or RankSign for rank metric, and the best practical for blindness is retrieving a small weight vector x of weight w_x from the syndrome Ax^T+Bx^T, for A and B random matrices. Hence we choose parameters according to these constraints. The size of the public key is only the size of public key of the signature since A and B are random and can be obtained from a small seed. At last the size of the signature the public key is the size of the proof of knowledge obtained from the concatenated Stern protocol, in our case the length of a vector of the ambient space is large compared to the size of a hash, hence on the average $4/3(n+n')$ per round, to be multiplied by a value l such that $(2/3)^l = 2^\lambda$ for λ a security parameter. For rank metric if one considers vectors over $GF(q^m)$ the size of vectors in bits is $(n+n')m$ and the average size of data for one round is also $4/3(n+n')m$.

B. Parameters

We now give example of parameters for our scheme, considering parameters for which a word of weight w_x is unique with very strong probability:

$\diamond$ Hamming distance:

We consider the parallel CFS signature scheme with parameters $n = 2^{18}$, $w_y = 9$ and $k = 162$, $n' = 6000$, $k' = 300$ and $w_x = 30$. For that case the security of parallel CFS is 2^{82} and 2^{91} for the cost of recovering a unique (with strong probability) x of weight 30 from its syndromes by matrices A and B.

$\diamond$ rank metric:

We consider the RankSign signature scheme for $n = 23$, over $GF(q^{24})$ and $q = 2^8$ (see [13]), with $k = 10, n' = 28, k' = 10$, $rank(x) = 5$ and $rank(y) = 8$. Notice that RankSign is very fast protocol, so that the signature can be computed easily. We choose parameters so that x is unique with a strong probability. We give in Table 1 the different sizes of keys (in bytes) and signature we obtain with previous parameters, our parameters are not as good as the best parameters for number theory [16] but are designed in a quantum resistant setting.

Metric	pk size	signature size	security bits
Hamming distance (CFS)	3MB	3.1MB	82
Rank metric (RankSign)	15kB	200kB	100

TABLE I
Examples of parameters for Hamming and rank metrics

IX. Conclusion

In this paper we propose the first blind signature algorithm for coding theory. Our approach is completely new for coding theory, our scheme is generic for a given trapdoor function $D_H()$. Our protocol is based on a concatenated version of the Stern protocol. We give examples of parameters for Hamming distance and rank metric, these parameters are practical for rank metric and a little less practical for Hamming distance.

References

[1] Q. Alamélou, O. Blazy, S. Cauchie, and P. Gaborit. A code-based group signature scheme. In *Des. Codes Cryptography 82(1-2)*, pages 469–493, 2017.

[2] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A practical group signature scheme based on rank metric. In *Arithmetic of Finite Fields - 6th International Workshop, WAIFI 2016, Ghent, Belgium, July 13-15, 2016, Revised Selected Papers*, pages 258–275, 2016.

[3] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography with constant input locality. In *CRYPTO 2007, volume 4622 of* LNCS*, pages 92–110.*

[4] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology - CRYPTO 1982*, LNCS, pages 199–203. Springer, 1982.

[5] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *ASIACRYPT 2001*, pages 157–174, 2001.

[6] J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *ITW 2011*, pages 282–286, Paraty, Brazil, October 2011.

[7] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86*, pages 186–194.

[8] M. Finiasz. Parallel-CFS: Strengthening the CFS McEliece-based signature scheme. In *SAC*, pages 159–170, 2010.

[9] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In *CRYPTO 2006,*, pages 60–77.

[10] P. Gaborit, A. Hauteville, and J.-P. Tillich. Ranksynd a PRNG based on rank metric. In *proceedings of PQCrypto 2016*, pages 18–28, 2016.

[11] P. Gaborit, O. Ruatta, and J. Schrek. On the complexity of the rank syndrome decoding problem. *Information Theory, IEEE Transactions on*, 62(2):1006–1019, 2016.

[12] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor. New results for rank-based cryptography. In *AFRICACRYPT 2014*, pages 1–12, 2014.

[13] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor. Ranksign: An efficient signature algorithm based on the rank metric. In *Post-Quantum Cryptography*, pages 88–107, 2014.

[14] P. Gaborit, J. Schrek, and G. Zémor. Full cryptanalysis of the chen identification protocol. In *Post-Quantum Cryptography 2011, p. 35-50.*

[15] P. Gaborit and G. Zémor. On the hardness of the decoding and the minimum distance problems for rank codes. *IEEE Trans. Information Theory 62(12): 7245-7252 (2016).*

[16] Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In *EUROCRYPT 2014*, pages 477–495, 2014.

[17] Gregory Landais and Nicolas Sendrier. Implementing CFS. In *INDOCRYPT 2012,*, pages 474–488, 2012.

[18] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[19] Markus Rückert. Lattice-based blind signatures. In *ASIACRYPT*, pages 413–430, 2010.

[20] J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, November 1996.

[21] Jacques Stern. A new identification scheme based on syndrome decoding. In *CRYPTO93*, pages 13–21, 1994.