A new one-time signature scheme from syndrome decoding

Paulo S. L. M. Barreto[1]* and Rafael Misoczki[1]

Departamento de Engenharia de Computação e Sistemas Digitais (PCS),
Escola Politécnica, Universidade de São Paulo, Brazil.
{pbarreto,rmisoczki}@larc.usp.br

Abstract. We describe a one-time signature scheme based on the hardness of the syndrome decoding problem, and prove it secure in the random oracle model. Our proposal can be instantiated on general linear error correcting codes, rather than restricted families like alternant codes for which a decoding trapdoor is known to exist.

1 Introduction

Digital signature algorithms are among the most useful and recurring cryptographic schemes. It is thus of utmost importance to ensure that suitable, provably secure post-quantum signature schemes are available for deployment, should quantum computers become a technological reality.

The Courtois-Finiasz-Sendrier (CFS) signature scheme [5] is one of the most successful and sports a formal security analysis, but it must be instantiated on top of codes for which an efficient decoder is known (and thus has to be disguised *a priori*), and moreover must have a high density of decodable syndromes, which in practice means only binary Goppa codes are suitable. The Kabatianskii-Krouk-Smeets (KKS) one-time signature scheme [10], on the other hand, can be instantiated on top of general codes, but it lacks a formal security analysis.

Our contribution in this paper is a syndrome-based one-time signature scheme that:

- admits a proof of EUF-1CMA security in the random oracle model;
- uses generic codes for which no efficient decoder is known, rather than restricted families where such trapdoors are known to exist.

The remainder of this paper is organized as follows. Section 2 discusses theoretical preliminaries for the presentation and analysis of our proposal.

* Supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under research productivity grant 312005/2006-7 and universal grant 485317/2007-9.

Section 3 describes the proposed signature scheme. Section 4 formally analyzes the proposal and presents a security proof in the random oracle model. Section 5 discusses parameter selection in practical scenarios. We conclude in Section 6.

2 Preliminaries

We now recapitulate some essential concepts from coding theory and security notions for signature schemes.

A binary linear error-correcting code of length n and rank (or dimension) k, or $[n, k]$-code for short, is a linear subspace of $\mathbb{F}_2^n$ of dimension k. If its minimum distance is d, it is called an $[n, k, d]$-code. An $[n, k]$-code $\mathcal{C}$ is specified be either a generator matrix $G \in \mathbb{F}_2^{k \times n}$ or by a parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ as $\mathcal{C} = \{mG \in \mathbb{F}_2^n \mid m \in \mathbb{F}_2^k\} = \{c \in \mathbb{F}_2^n \mid Hc^{\mathrm{T}} = 0\}$.

The syndrome decoding problem, as well as the closely-related general decoding problem, are classical in coding theory and known to be NP-complete [1]:

Definition 1 (Syndrome decoding problem). *Let r, n, and w be integers, and let (H, w, s) be a triple consisting of a matrix $H \in \mathbb{F}_2^{r \times n}$, an integer $w < n$, and a vector $s \in \mathbb{F}_2^r$. Does there exist a vector $e \in \mathbb{F}_2^n$ of weight $\mathsf{wt}(e) \leqslant w$ such that $He^T = s^T$?*

Definition 2 (General decoding problem). *Let k, n, and w be integers, and let (G, w, s) be a triple consisting of a matrix $G \in \mathbb{F}_2^{k \times n}$, an integer $w < n$, and a vector $c \in \mathbb{F}_2^n$. Does there exist a vector $m \in \mathbb{F}_2^k$ such that $\mathsf{wt}(mG + c) \leqslant w$?*

We write SDP(n, r, w) for the syndrome decoding problem with parameters as stipulated in the above definitions, and similarly GDP(n, k, w) for the general decoding problem. For convenience we also define the ℓ-SDP(n, r, w) and the ℓ-GDP(n, k, w) to consist of solving ℓ simultaneous instances of the SDP(n, r, w) or the GDP(n, k, w), respectively.

We now provide a quantitative definition of the hardness of the search version of the SDP(n, r, w).

Definition 3 (Computational syndrome decoding). *A probabilistic algorithm $\mathcal{D}$ is said to (τ, ε)-break (the search version of) the SDP(n, r, w) for an $[n, n - r]$-code if $\mathcal{D}$ runs in at most τ steps and decodes a syndrome $s^T = He^T$ into an error vector e of weight $\mathsf{wt}(e) \leqslant w$ given the input $H \in \mathbb{F}_2^{r \times n}$, w, and s with probability at least ε, where the probability is taken over the coins $\mathcal{S}$ tosses and e is uniformly sampled from $\mathbb{F}_2^n$ with $\mathsf{wt}(e) \leqslant w$.*

All $[n,k,d]$ codes satisfy the Singleton bound, which states that $d \leqslant n-k+1$. A binary linear $[n,k,d]$ code is ensured to exist as long as:

$$\sum_{j=0}^{d-2} \binom{n-1}{j} < 2^{n-k}.$$

This is called the Gilbert-Varshamov (GV) bound. Random binary codes are known to meet the GV bound, in the sense that the above inequality comes very close to being an equality [6]. No family of binary codes is known that can be decoded in subexponential time up to the GV bound, nor is any subexponential algorithm known that can decode general codes up to the GV bound.

Definition 4. *A* signature scheme *is a triple (Keygen, Sign, Verify) consisting of the following algorithms:*

- *The probabilistic key pair generation algorithm Keygen, given as input a security parameter 1^λ, outputs a pair (sk, pk) consisting of a private signing key sk and a matching public verification key pk.*
- *The signing algorithm Sign, given as input a key pair (sk, pk) generated by Keygen and a message m, produces a signature σ.*
- *The verification algorithm Verify, given as input a public key pk, a signed message m and its signature σ, outputs either valid or invalid with the property that if $(sk, pk) \leftarrow \mathsf{Keygen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{Sign}(sk, pk, m)$, then $\mathsf{Verify}(pk, m, \sigma) = \mathsf{valid}$.*

The strongest security notion for one-time signatures is existential unforgeability against one-chosen-message attacks (EUF-1CMA), whereby an attacker cannot fake a signature based on the public key alone [9].

Definition 5 (EUF-1CMA security). *A probabilistic algorithm $\mathcal{A}$ is said to $(\tau, q_H, 1, \varepsilon)$-break a signature scheme if, after running for at most τ steps, making at most q_H adaptive queries to a hash function oracle, and at most one query to a signing oracle for the signature of a message m of its choice, $\mathcal{A}$ outputs a forged signature σ on some other message $m' \neq m$ with probability at least ε, where the probability is taken over the coins $\mathcal{A}$ tosses, the Keygen and Sign algorithms, and the hash function oracle. A signature scheme is then said to be $(\tau, q_H, 1, \varepsilon)$-secure, or EUF-1CMA for short, if no adversary $\mathcal{A}$ can $(\tau, q_H, 1, \varepsilon)$-break it.*

3 Proposed signature scheme

Our proposal is inspired by both Schnorr signatures [13] based on the discrete logarithm problem, and KKS signatures [10] based on the syndrome decoding problem. We use a random oracle $\mathbf{h} : \{0,1\}^* \times \mathbb{F}_2^r \to \mathbb{F}_2^k \setminus \{0\}$.

Notation: $x \overset{\$}{\leftarrow} U$ means that variable x is uniformly chosen at random from the set U. Given a matrix $H \in \mathbb{F}_2^{r \times n}$ and a set $J \in 2^{\{1 \dots n\}}$ of cardinality m, $H(J) \in \mathbb{F}_2^{r \times m}$ denotes the matrix obtained from H by keeping the columns indicated in J and deleting the rest.

- Keygen: Given a security parameter λ, choose suitable integers k, n, r, u, w such that the actual difficulty of the SDP(n, r, u) meets the level 2^λ, with $u \leqslant w$.
  The private key is a generator matrix $P \overset{\$}{\leftarrow} \mathbb{F}_2^{k \times n}$ of a random $[n, k]$-code whose codewords have weight not exceeding w. This weight limit holds in particular for the rows of P. If each bit out of the w bound is chosen uniformly from $\mathbb{F}_2$, the weight of each row of P follows, by the central limit theorem, a normal distribution with mean $w/2$ and standard deviation $\sqrt{w}/2$. In practice we ask that the weight of any row of P be close to $w/2$ (within, say, $3\sqrt{w}/2$, as is the case of about 99.7% of all random rows by the 3σ rule).
  The public key is a pair (H, V) where $H \in \mathbb{F}_2^{r \times n}$ is a parity-check matrix of an $[n, n-r, d \geqslant 4w+1]$-code and $V \leftarrow HP^{\mathrm{T}} \in \mathbb{F}_2^{r \times k}$.
  One can see that directly recovering P from H and V alone amounts to solving an instance of the k-SDP(n, r, u) with $|u - w/2| \leqslant 3\sqrt{w}/2$.
- Sign: To sign a message $m \in \{0,1\}^*$ under the private key $P \in \mathbb{F}_2^{k \times n}$, the signer computes the following:

$$\begin{aligned} e &\overset{\$}{\leftarrow} \mathbb{F}_2^n \text{ such that } \mathsf{wt}(e) = w \\ s^{\mathrm{T}} &\leftarrow He^{\mathrm{T}} \\ h &\leftarrow \mathbf{h}(m, s) \\ c &\leftarrow hP + e \end{aligned}$$

  The signature is the pair $(h, c) \in \mathbb{F}_2^k \times \mathbb{F}_2^n$. Since the maximum weight of the code generated by P is w, clearly $\mathsf{wt}(hP + e) \leqslant \max_h \mathsf{wt}(hP) + \mathsf{wt}(e) = 2w$, and hence legitimate signatures satisfy $\mathsf{wt}(c) \leqslant 2w$. Naively, a signature (h, c) occupies $k + n$ bits, but the weight restriction on c suggests a more compact representation by its rank in some conventional ordering (e.g. colex), i.e. $\lg \binom{n}{2w}$ bits, plus the indication of the actual weight $\mathsf{wt}(c)$, yielding a total of $k + \lg \binom{n}{2w} + \lg(2w)$ bits per signature.

We point out that the actual weight of e could have been defined independently from the maximum weight of the code generated by P, but the security and practical considerations below suggest that this simple choice is close to optimal.

- Verify: To verify a signature $(h, c) \in \mathbb{F}_2^k \times \mathbb{F}_2^n$ for a message $m \in \{0,1\}^*$ under the public key $(H, V) \in \mathbb{F}_2^{r\times n} \times \mathbb{F}_2^{r\times k}$, the verifier checks that $\mathsf{wt}(c) \leqslant 2w$, computes $s^{\mathrm{T}} \leftarrow Hc^{\mathrm{T}} + Vh^{\mathrm{T}}$, $v \leftarrow \mathbf{h}(m, s)$, and accepts iff $v = h$.

The consistency of this scheme for legitimate signatures is established by the fact that, by the definition of c, s, and V, $Hc^{\mathrm{T}} = H(hP + e)^{\mathrm{T}} = HP^{\mathrm{T}}h^{\mathrm{T}} + He^{\mathrm{T}} = Vh^{\mathrm{T}} + s^{\mathrm{T}}$. Hence, $s^{\mathrm{T}} = Hc^{\mathrm{T}} + Vh^{\mathrm{T}}$ as expected, so it necessarily follows that $v = \mathbf{h}(m, s) = h$.

4 Security analysis

We begin by showing that our proposal cannot be turned to a multisigning scheme. We proceed to show that, as a one-time scheme, it is EUF-1CMA secure in the random oracle model.

4.1 The impossibility of multisigning

The condition that all words of the code generated by P have weight bound by w poses a very strict constraint on the density of P, by virtue of the following property:

Theorem 1. *Let $\mathcal{C}$ be a random binary $[n', k]$-code in systematic form. Let $0 < \delta < 1$ and $r' = n' - k$. Then:*

$$\Pr\left[\forall v \in \mathcal{C} : \frac{n'}{2}(1-\delta) \leqslant \mathsf{wt}(v) \leqslant \frac{n'}{2}(1+\delta)\right] \geqslant 1 - 2^{-r'+n'H_2(\delta)+1}$$

where $H_2(x) = -x \lg x - (1-x)\lg(1-x)$ is the binary entropy function.

Proof. See [4, Proposition 3]. □

Due to Theorem 1, a completely random code of length n would display $w = (\frac{n}{2})(1+\delta)$ for some $0 < \delta < 1$ with high probability, but this is incompatible with the requirement that the code generated by $H \in \mathbb{F}^{r\times n}$, also of length n, have minimum distance at least $d \geqslant 4w + 1 = 2n(1+\delta) + 1 > n$, which is clearly impossible.

Therefore we are forced to choose a very sparse P instead. However, this means that all but a set $J \subset \{1 \dots n\}$ of $n_0 = \#J$ columns of P are null, for some n_0. By the above reasoning, $d \geqslant 2n_0(1+\delta)+1$, and by virtue of the Singleton bound $d \leqslant r+1$ so that $n_0 \leqslant \frac{r}{2(1+\delta)} < r$. Thus, an adversary who knows J could solve the overdetermined linear system $H(J)P(J)^{\mathrm{T}} = V$ to recover $P(J)$ and hence P.

We now show how to recover J from a collection of ℓ valid signatures, improving and extending a technique put forward in [4]. Assume initially a scenario where the error vector is null, so that the c component of each signature has the form $c = hP$ (this corresponds to the KKS setting). Each nonzero column of c reveals one column of $P(J)$ and hence one element of J, since the null columns will always yield a zero column in c. Since h is the output of a random oracle and thus uniformly distributed, each column of $hP(J)$ is nonzero with probability $1/2$, and hence each signature reveals on average half of the still unknown elements of J. Therefore about $\lg n_0$ signatures are expected to reveal the whole J. At this point one could continue an information-theoretical attack to recover $P(J)$ as suggested in [4], but the simpler method above of solving the overdetermined linear system $H(J)P(J)^{\mathrm{T}} = V$ for $P(J)$ yields the solution without the need for any further signatures.

To tackle the noise introduced by the error vector, we resort to a counting procedure. Each column in $c = hP+e$ receives a contribution from hP with probability $\Pr[1] = 1/2$ as already pointed out, and a contribution from e with probability $\Pr[1] = w/n$. By the central limit theorem, the sum of ℓ independent binary variables (lifted to $\mathbb{Z}$) that are randomly sampled with $\Pr[1] = \delta$ has mean $\ell\delta$ and standard deviation $\sqrt{\ell\delta(1-\delta)}$. Thus the number of times a column of c corresponding to a nontrivial column in P (i.e. an element of J) assumes the value 1 gets an average contribution $\mu_0 \approx \ell/2$ with standard deviation $\sigma_0 \approx \sqrt{\ell}/2$ from hP, and an average contribution $\mu_e \approx \ell w/n$ with standard deviation $\sigma_e \approx \sqrt{\ell(w/n)(1-w/n)}$ from e. It is also necessary to take into account that the two distributions interfere with each other on the columns indicated by J. To distinguish the contributions, one needs to set ℓ so that the difference between the lower count due to hP and the upper count due to e (i.e. the actual count on the columns indicated by J) exceeds the upper count due to e (i.e. the count on columns outside J, which is due purely to e), say, $(\mu_0 - m_0\sigma_0) - (\mu_e + m_0\sigma_e) > \mu_e + m_0\sigma_e$ for a number m_0 of standard deviations. Therefore $(\ell/2 - m_0\sqrt{\ell}/2) > 2\ell w/n + 2m_0\sqrt{\ell(w/n)(1-w/n)}$, or

$$\ell \geqslant m_0^2 \left[\frac{1+4\sqrt{(w/n)(1-w/n)}}{1-4w/n}\right]^2 .$$

Notice that this reasoning includes the case where e is null, whereby $w = 0$ and the condition $\mu_0 - m_0\sigma_0 > 0$ leads to $\ell \geqslant m_0^2$. Of course, the probability of these conditions being satisfied is that of the count population lying within a range of m_0 standard deviations from the mean, e.g. setting $m_0 = 3$ reveals J with probability about 99.7% by the 3σ rule. Since the null error situation asks equivalently for $\ell \geqslant \lg n_0$ and $\ell \geqslant m_0^2$, it is natural to set $m_0^2 \approx \lg n_0$, or

$$\ell \geqslant \left[\frac{1 + 4\sqrt{(w/n)(1 - w/n)}}{1 - 4w/n}\right]^2 \lg n_0.$$

On the constructive side these observations suggest how one could obtain a suitable P for a one-time (or few-times at best) signature scheme, namely, choose a random $J \subset \{1 \dots n\}$ with $\#J = n_0$, and take a code generated by a random matrix $P_0 \in \mathbb{F}_2^{k \times n_0}$ and embed it into a longer code generated by $P \in \mathbb{F}_2^{k \times n}$ such that $P(J) = P_0$. One must take care to make the number of possible sets J high enough, i.e. $\binom{n}{n_0} \geqslant 2^\lambda$ for the adopted security parameter λ. Our default suggestion is to take $n_0 = w$. For the parameters suggested on Table 1 one has $\ell \approx 3.5 \lg w$.

4.2 EUF-1CMA security

Given a message m, the Pointcheval-Stern generic digital signature scheme [12] produces triples (σ_1, h, σ_2) where σ_1 is randomly sampled from a large set, h is the hash value of (m, σ_1), and σ_2 only depends on σ_1, the message m, and h. We write $(m, \sigma_1, h, \sigma_2)$ for the resulting signature on message m.

We argue that our proposal meets the definition of a Pointcheval-Stern generic signature scheme. With the notation in Section 3, the triples are (s, h, c). Even though the signing algorithm properly yields only the pair (h, c), s can be readily obtained from it, as is clear from the verification algorithm. Component s is clearly sampled from a large set of size $\binom{n}{w}$, and component h is indeed the hash value of (m, s). It remains to show that the c component depends only on s, m, and h. We first notice that, although c is directly dependent on e rather than on its syndrome s, there is a unique e of weight w for a given valid s because the minimum distance of the code defined by H is $4w+1 > 2w+1$. Hence the explicit dependence on e in the relation $c = hP + e$ reflects an implicit but unambiguous dependence on s. Now assume there were a distinct but valid triple (s, h, c') with $c' \neq c$. This would mean $Hc'^{\mathrm{T}} = Vh^{\mathrm{T}} + s^{\mathrm{T}} = Hc^{\mathrm{T}}$ and hence $H(c + c')^{\mathrm{T}} = 0$, i.e. $c + c'$ is a codeword of the code defined by H. But this is impossible, because $\mathsf{wt}(c+c') \leqslant 2 \times 2w$ and the weight of any nonzero codeword of H is at least $4w+1$. Therefore, c is uniquely determined by, and does indeed depend only on, s, m, and d. This is in fact the rationale for the required minimum distance of H.

The following well known result, the Forking Lemma (in its restricted and general forms), is central to the security assessment of the proposed scheme:

Theorem 2 (The Restricted Forking Lemma). *Let (Keygen, Sign, Verify) be a generic digital signature scheme with security parameter λ. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by q_H the number of queries that $\mathcal{A}$ can ask to the random oracle. Assume that, within time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geqslant 7q_H/2^\lambda$, a valid signature $(m, \sigma_1, h, \sigma_2)$. Then there is another machine which has control over $\mathcal{A}$ and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$, in expected time $T' \leqslant 84480 q_H T/\varepsilon$.*

Proof. See [12, Theorem 1]. □

Corollary 1. *In the conditions of the Restricted Forking Lemma, for any given ℓ there is a machine $\mathcal{A}_\ell$ that can produce ℓ valid signatures $(m, \sigma_1, h_j, \sigma_{2,j})$, $j = 1 \ldots \ell$, such that the h_j are all distinct, in expected time $T' \leqslant 84480 \ell q_H T/\varepsilon$.*

Proof. It suffices to iterate the Restricted Forking Lemma using a family of ℓ distinct random oracles to produce the h_j for $j = 1 \ldots \ell$. □

Theorem 3 (The General Forking Lemma). *Let (Keygen, Sign, Verify) be a generic digital signature scheme with security parameter λ. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by q_H and q_S the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geqslant 10(q_S+1)(q_S+q_H)/2^\lambda$, a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{A}$ replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time $T' \leqslant 120686 q_H T/\varepsilon$.*

Proof. See [12, Theorem 3]. □

Corollary 2. *In the conditions of the General Forking Lemma, for any $\ell > 0$ there is a machine $\mathcal{A}_\ell$ that can produce ℓ valid signatures $(m, \sigma_1, h_j, \sigma_2^{(j)})$, $j = 1 \dots \ell$, such that the h_j are all distinct, in expected time $T' \leqslant 120686 \ell q_H T/\varepsilon$.*

Proof. It suffices to iterate the General Forking Lemma using a family of ℓ distinct random oracles to produce the h_j for $j = 1 \dots \ell$. □

The following theorem establishes that the proposed scheme is secure against attacks where the adversary can query the hash oracle but not the signer.

Theorem 4. *Assume that, within a time bound T, an attacker $\mathcal{A}$ performs an existential forgery under a no-message attack against the proposed signature scheme, with probability $\varepsilon \geqslant 7q_H/2^\lambda$ where q_H denotes the number of queries that $\mathcal{A}$ can ask to the random oracle. Then the k-SDP(n, r, w) can be solved in expected time $T' \leqslant 84480 \ell q_H T/\varepsilon$ where $\ell = O(\lambda)$.*

Proof. From Corollary 1, after $\ell = O(\lg w) = O(\lambda)$ polynomial replays of the attacker $\mathcal{A}$, we obtain ℓ valid signatures (m, s, h_j, c_j), $j = 1 \dots \ell$, where the h_j, and hence also the c_j, are all distinct. Now it suffices to apply to this collection the procedure outlined in Section 4.1 to recover J and hence P. □

To finally establish EUF-1CMA security, we need the following indistinguishability result:

Theorem 5. *The triples (s, h, c) of the proposed scheme can be simulated without knowing the private key P, in the sense of being indistinguishable from legitimate triples unless the adversary is able to solve the SDP(n, r, w).*

Proof. A valid triple (s, h, c), either legitimate or simulated, must satisfy $\mathsf{wt}(c) \leqslant 2w$ and $s^{\mathrm{T}} = Hc^{\mathrm{T}} + Vh^{\mathrm{T}}$. Clearly, one can simulate triples without knowing the private key P by simply choosing $h \xleftarrow{\$} \mathbb{F}_2^k - \{0\}$, $c \xleftarrow{\$} \mathbb{F}_2^n$ such that $\mathsf{wt}(c) \leqslant 2w$, and $s^{\mathrm{T}} \leftarrow Hc^{\mathrm{T}} + Vh^{\mathrm{T}}$.

Let $Q \neq P$ be a solution of the linear system $HQ^{\mathrm{T}} = V$, so that $Q = W + P$ for some nonzero codeword array W of the code defined by H. The c component of any valid triple (s, h, c) can always be written as $c = hQ + e$ for some e, since one can set $e \leftarrow c + hQ$. In that case, $He^{\mathrm{T}} = Hc^{\mathrm{T}} + (HQ^{\mathrm{T}})h^{\mathrm{T}} = s^{\mathrm{T}}$. However, a valid triple must satisfy the weight requirement $\mathsf{wt}(hQ + e) \leqslant 2w$. For $Q \neq P$, $\mathsf{wt}(hQ) \geqslant |\mathsf{wt}(hW) -$

$\mathsf{wt}(hP)| \geqslant |4w+1-w| = 3w+1$, and hence $\mathsf{wt}(hQ+e) \geqslant |3w+1-\mathsf{wt}(e)|$. The only way this does not exceed the upper bound of $2w$ is by imposing $\mathsf{wt}(e) \geqslant w+1$. Thus, all legitimate triples (s, h, c) correspond to $s^{\mathrm{T}} = He^{\mathrm{T}}$ with $\mathsf{wt}(e) \leqslant w$, whereas any simulated triple corresponds to $s^{\mathrm{T}} = He^{\mathrm{T}}$ with $\mathsf{wt}(e) > w$. Therefore, telling a simulated triple from a legitimate one is the same as solving the $\mathrm{SDP}(n, r, w)$. Notice that the circumstance where $Q = P$ as a solution of the linear system $HQ^{\mathrm{T}} = V$ can be detected by the fact that $\mathsf{wt}(hP) \leqslant w$ whereas $\mathsf{wt}(hQ) \geqslant 3w + 1$ for $Q \neq P$. Needless to say, this case is not considered because of the assumption that the simulator does not know the private key. □

We are now in a position to prove the security of our proposal as a one-time signature scheme.

Theorem 6. *Let $\mathcal{A}$ be an attacker which performs, within a time bound T, an existential forgery under a one-chosen-message attack against the proposed signature scheme with probability $\varepsilon \geqslant 20(1 + q_H)/2^{\lambda}$ where q_H denotes the number of queries that $\mathcal{A}$ can ask to the random oracle. Then the k-SDP(r, n, w) can be solved within expected time $T' \leqslant 120686\ell q_H T/\varepsilon$ where $\ell = O(\lambda)$.*

Proof. From Corollary 2, after $\ell = O(\lg w) = O(\lambda)$ polynomial replays of the attacker $\mathcal{A}$, we obtain ℓ valid signatures (m, s, h_j, c_j), $j = 1 \dots \ell$, where the h_j, and hence also the c_j, are all distinct. Now it suffices to apply to this collection the procedure outlined in Section 4.1 to recover J and hence P. □

5 Choosing parameters

Table 1 suggests parameters for practical security levels. Parameter w is chosen so that the effort of exhaustively guessing which w of the set bits of the c component in a signature correspond to the error vector e (so that the remaining set bits would reveal partial information on J). The size of J is $\lceil \lg \binom{n}{w} \rceil$.

We choose r to be such that 2 is a primitive element in $\mathbb{F}_r$, so that almost all double-circulant $r \times 2r$ parity-check matrices H define a code meeting the GV bound [7]. Settings where H is double-dyadic are similarly possible. Unfortunately these techniques cannot be used for P, since the structure would become apparent in the c component of legitimate signatures and greatly reduce the effort to recover J.

The size $|(h, c)|$ of each signature is at most $k + \lg \binom{n}{2w} + \lg(2w)$, representing c as its rank in some conventional ordering of binary strings and its weight. The size of a signature in a Merkle tree scheme capable of yielding up to $2^{\lambda/2}$ signatures is also shown.

Table 1. Suggested parameters for standard security levels.

λ	k	$w = \|P_0\|$	$\|J\|$	$r = \|H\|$	n	$\|V\|$	SDP$_H$	$\|(h, c)\|$	Merkle tree
80	160	170	1118	3083	6166	493280	82–120	2062	504825
112	224	238	1569	4349	8698	974176	113–158	2891	993960
128	256	272	1791	4933	9866	1262848	128–177	3298	1287463
192	384	408	2690	7411	14822	2845824	192–252	4946	2895045
256	512	544	3588	9883	19766	5060096	256–326	6594	5142109

Table 2 compares some code-based signature schemes, all at the 2^{80} security level.

Table 2. Comparing coding-based signature schemes.

scheme	$\|sk\|$	$\|pk\|$	sig bits	signing time	code	sig/key	sec proof?
CFS	444434	5898240	180	$O(t!\,tn)$	trapdoor	$2^{O(n)}$	yes
Stern[†]	694	347	$\sim$ 120000	$O(n^2 \lg n)$	generic	$2^{O(n)}$	yes
KKS[‡]	2726	176900	1942	$O(n^2 \lg n)$	generic	$O(1)$	no
Ours	1288	496363	2062	$O(n^2 \lg n)$	generic	$O(1)$	yes

[†] Quasi-cyclic setting [7], assuming $O(n)$ Fiat-Shamir rounds.

[‡] KKS-3 version #2 [4, Table 4].

6 Conclusion

We have described a signature scheme whose security stems from the hardness of the syndrome decoding problem, and showed it to be EUF-1CMA secure in the random oracle model. The proposed algorithm uses general codes rather than restricted families for which a decoding algorithm is known.

It would be desirable to modify the scheme so as to obtain a security proof without resorting to random oracles, for instance, by trying to replace each such occurrence by a uniformly sampled member of a family of universal one-way hash functions. Currently known proof techniques seem to impose considerable difficulties to achieve this goal. We point out, however, that using the scheme in a Merkle tree setting would make the use of random oracles almost unavoidable.

A drawback of the proposed method is its reliance upon the basic Forking Lemma. Directly programming the oracle in a tailored security reduction might lead to tighter requirements and smaller keys, possibly matching the KKS scheme. We leave this question as an open problem for further research.

Acknowledgements

We are most grateful to Pierre-Louis Cayrel, Eike Kiltz, and Benoît Libert for their comments during the preparation of this work.

References

1. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
2. S. A. Brands. Untraceable off-line cash in wallets with observers. In *International Cryptology Conference on Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318, New York, NY, USA, 1994. Springer New York.
3. P.-L. Cayrel, P. Gaborit, D. Galindo, and M. Girault. Improved identity-based identification using correcting codes. preprint, 2009. arXiv:0903.0069v1 [cs.CR].
4. P.-L. Cayrel, A. Otmani, and D. Vergnaud. On Kabatianskii-Krouk-Smeets signatures. In *International Workshop on the Arithmetic of Finite Fields – WAIFI'2007*, volume 4547 of *Lecture Notes in Computer Science*, page 237. Springer, 2007.
5. N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology – Asiacrypt'2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174, Gold Coast, Australia, 2001. Springer.
6. J. MacWilliams F and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland Amsterdam, 1978.
7. P. Gaborit and M. Girault. Lightweight code-based identification and signature. In *Proceedings of the IEEE International Symposium on Information Theory – ISIT'2007*, volume 7. IEEE, 2007.
8. D. Galindo and F. D. Garcia. A Schnorr-like lightweight identity-based signature scheme. In *Progress in Cryptology – AFRICACRYPT'2009*, volume 5580 of *Lecture Notes in Computer Science*, pages 135–148, New York, NY, USA, 2009. Springer Berlin / Heidelberg.
9. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 1988.
10. G. Kabatianskii, E. Krouk, and B. Smeets. A digital signature scheme based on random error-correcting codes. In *IMA International Conference on Crytography and Coding*, Lecture Notes in Computer Science, pages 161–167. Springer, 1997.
11. R. Misoczki and P. S. L. M. Barreto. Compact mceliece keys from goppa codes. In *Selected Areas in Cryptography – SAC'2009*, Lectures Notes in Computer Science. Springer, 2009. to appear.

12. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
13. C. P. Schnorr. Efficient identification and signatures for smart cards. In *International Cryptology Conference on Advances in Cryptology – CRYPTO'89*, Lecture Notes in Computer Science, pages 239–252, New York, NY, USA, 1989. Springer New York, Inc.

A Counterexamples (*aka* broken schemes)

The Schnorr signature scheme has been used in a number of derived protocols. It is conceptually possible to adapt such derivatives to the syndrome-based setting we propose, but there is no guarantee that they remain secure when transplanted to the syndrome-based setting. An intriguing counterexample is a syndrome-based variant of the Brands blind signature scheme [2]. The operation proceeds as follows.

- Commit: The blind signer samples a uniformly random $e \xleftarrow{\$} \mathbb{F}_2^n$ of weight $\mathsf{wt}(e) \leqslant w$, computes its syndrome $s^{\mathrm{T}} \leftarrow He^{\mathrm{T}}$ and sends it to the user who requested a blind signature.
- Challenge: The user who wants to obtain a blind signature for a message m chooses two random vectors $\eta \xleftarrow{\$} \mathbb{F}_2^k$ and $\gamma \xleftarrow{\$} \mathbb{F}_2^n$ such that $\mathsf{wt}(\gamma) \leqslant w$, blinds the received syndrome as $s'^{\mathrm{T}} \leftarrow s^{\mathrm{T}} + H\gamma^{\mathrm{T}} + V\eta^{\mathrm{T}}$, computes $h' \leftarrow \mathbf{h}(m, s')$ and sends $h \leftarrow h' + \eta$ to the blind signer.
- Response: The blind signer computes $c \leftarrow hP + e$ and sends c back to the user.
- Extract: The user computes $c' \leftarrow c + \gamma$ and sets the pair (h', c') as the signature of m. Notice that $s'^{\mathrm{T}} = Hc'^{\mathrm{T}} + Vh'^{\mathrm{T}}$ and $\mathbf{h}(m, s') = h'^{\mathrm{T}}$ as expected, and $\mathsf{wt}(c') \leqslant 3w$, which is the modified weight condition for this scheme.

The problem with this protocol is that, given a signature (s', h', c'), a blind signer who keeps track of who asked for each particular (s, h, c) can find a triple (s, h, c) such that $\mathsf{wt}(c + c') \leqslant w$ and check that $(s + s')^{\mathrm{T}} = H(c+c')^{\mathrm{T}} + V(h+h')^{\mathrm{T}}$, thus discovering that user's identity. Therefore this scheme is non-anonymous and hence broken. The Okamoto-Schnorr blind signature scheme is somewhat more involved than the Brands scheme, but fails to be anonymous for the same reason.

Another counterexample is the Galindo-Garcia lightweight identity-based signature scheme [8], which can be formally made into a syndrome-based variant along the lines of our proposal. The scheme makes use of two random oracles $\mathbf{g} : \{0,1\}^* \times \mathbb{F}_2^r \times \{0,1\}^* \to \mathbb{F}_2^k$ and $\mathbf{h} : \{0,1\}^* \times \mathbb{F}_2^{k \times r} \to \mathbb{F}_2^{k \times k}$. The Keygen algorithm is used to generate the key pair for the trust authority. The remainder of the scheme consists of the following algorithms.

- $\mathsf{Extract}$: Given an identity $\mathsf{id} \in \{0,1\}^*$, the trust authority chooses a uniformly random matrix $E \xleftarrow{\$} \mathbb{F}_2^{k\times n}$ with rows of weight not exceeding w, then computes $S^{\mathrm{T}} \leftarrow HE^{\mathrm{T}} \in \mathbb{F}_2^{r\times k}$, $D \leftarrow \mathbf{h}(\mathsf{id}, S) \in \mathbb{F}_2^{k\times k}$, $C \leftarrow DP + E \in \mathbb{F}_2^{k\times n}$ such that C is the generator matrix of an $[n, k]$-code whose codewords have maximum weight w', and outputs the identity-based private key $(C, S) \in \mathbb{F}_2^{k\times n} \times \mathbb{F}_2^{k\times r}$. Strictly speaking S is public information, since it will accompany the signatures.
- Sign: To sign a message m, the user whose identity is id and whose identity-based private key is $(C, S) \in \mathbb{F}_2^{k\times n} \times \mathbb{F}_2^{k\times r}$ chooses a uniformly random error vector $a \xleftarrow{\$} \mathbb{F}_2^n$ such that $\mathsf{wt}(a) \leqslant w'$, computes its syndrome $u^{\mathrm{T}} \leftarrow Ha^{\mathrm{T}} \in \mathbb{F}_2^r$, then $q \leftarrow \mathbf{g}(\mathsf{id}, u, m) \in \mathbb{F}_2^k$, and finally $b \leftarrow qC + a \in \mathbb{F}_2^n$. The signature is the triple $(u, b, S) \in \mathbb{F}_2^r \times \mathbb{F}_2^n \times \mathbb{F}_2^{k\times r}$, where $\mathsf{wt}(b) \leqslant 2w'$. Since S is shared by all signatures each user generates, a trivial optimization is possible by publishing S once and for all before the first signature is generated.
- Verify: To verify a purported signature $(u, b, S) \in \mathbb{F}_2^r \times \mathbb{F}_2^n \times \mathbb{F}_2^{k\times r}$ for message m and identity id, the verifier checks whether $\mathsf{wt}(b) \leqslant 2w'$, and if so, computes $D \leftarrow \mathbf{h}(\mathsf{id}, S) \in \mathbb{F}_2^{k\times k}$, $q \leftarrow \mathbf{g}(\mathsf{id}, u, m) \in \mathbb{F}_2^k$, $w^{\mathrm{T}} \leftarrow Hb^{\mathrm{T}} \in \mathbb{F}_2^r$, and accepts the signature iff $w = u + q(S + DV^{\mathrm{T}})$ and the weight inequalities hold.

Unfortunately the C component of the identity-based private key (C, S) consists of k signatures generated under the same private key P, thus violating the one-time restriction and leaking enough information to reveal P.