A Step Towards QC Blind Signatures

R. Overbeck

overbeck[]terra.es

Abstract. In this paper we propose a conversion from signature schemes connected to coding theory into blind signature schemes. We give formal security reductions to combinatorial problems not connected to number theory. This is the first blind signature scheme which can not be broken by quantum computers via cryptanalyzing the underlying signature scheme employing Shor's algorithms. We thus present a step towards diversifying computational assumptions on which blind signatures can be based.
We achieve blind signatures by a different concept of blinding: Instead of blinding the message, we blind the public key, such that generating a (blind) signature for the blinded key requires the interaction of the holder of the original secret key. To verify the blind signature, the connection between the original and the blinded key is proven by a static ZK proof. The major ingredient for our conversion is the PKP protocol by Shamir.

Keywords: Post-Quantum Cryptography, blind signatures, codes, lattices, public key cryptography,

1 Introduction

Cryptography based on the theory of codes and lattices has received a wide attention in the last years. This is not only because of the interesting mathematical background but as well because of Shor's algorithm, which showed that in a world where quantum computers are assumed to exist, number theoretic cryptosystems are insecure. There exist several interesting (a)symmetric encryption, authentication and signature schemes on the basis of coding and lattice theory which currently resist a crytpanalysis by quantum computers (A few examples are the NTRU and GGH encryption/signature schemes or the HB-family protocols). However, besides by generic constructions, few other cryptographic primitives can be realized without relying on number theory. In the case of blind signatures the situation in the (hypothetical) quantum computer world is thus quite desperate, since all generic constructions involve trusted third parties or one-way-trapdoor-permutations which so far can be constructed on number-theoretic assumptions, only.

One could argue that trusted third parties are not that bad. However, as blind signatures are involved in voting schemes, it will be hard to find a trusted third party for governmental elections – whom would you trust not to unblind your vote? In the paper world, this would be like sending your vote in a signed envelope to a counting assistant.

Our Contribution We present a conversion to obtain a blind signature scheme from signature schemes based on syndrome decoding which we assume to be secure troughout the paper. Our conversion does not require the interaction with 3rd parties or number theoretic assumptions (see below). However, since our proposal involves static zero knowledge proofs it relies on random oracles and we can not hope to prove its security e.g. in the UC-framework or in the case of adaptive adversaries. Instead, we give evidence that the resulting blind signature scheme is secure against active adversaries as long as some instances of $\mathcal{NP}$-hard problems are hard to solve.

The reader should not expect a complete solution for a blind singature scheme proven secure against quantum computers but a step toward this direction. Besides problems that might occurr by the combination of two different cryptographic schemes, we can not prove that a quantum computer has no advantage over an classical computer in attacking our scheme. (Nevertheless all the building blocks of our scheme resist against the state of art in quantum cryptanalysis.) Further, we use a new definition of a blind signature scheme, which might appear out of date in comparison to the definitions currently used in the context of nubmer theoretic blind signature schemes. Anyway we want to start the discussion if QC blind signatures exist. Comments and hints how to cryptanalyze our conversion are most welcome.

Our conversion can be adapted for lattice based signature schemes generating short vectors as signatures as long as all signature vectors are of the same Eukledian norm (NTRUSign for example can be modified to work that way). However, for the ease of understanding, we omit a detailed description.

The general idea is to derive a blinded public key from a public key of a code (or lattice) based signature scheme. Since both keys are related, anyone able to generate signatures for the blinded public key should be able to generate signatures for the original public key and vice versa. However, the relation of a signature of the blinded public key to a signa-

ture of the original public key should not be computable for the signer but only for the blinder.

To blind the public key, we propose to use permuted kernels (or a variant based on isometric subcodes / sublattices). Lets assume that for any code based signature scheme, the public key can be given in form of a generator matrix. Indeed, after adding some vectors to the public generator matrix, the secret key holder will still be able to produce signatures for this new code. However, by applying an isometry to this new code, we can prevent the secret key holder to identify the added vectors and thus the isometric subcode connection between the original code and the newly generated code. In order to keep the problem to generate signatures hard for the blinder, we will not allow him to add a unlimited number of vectors of his choice to the public generator matrix. Instead, we force him to do his choice among those in a vectorspace generated by a pseudo random number generator.

So far our blind signature scheme works as follows: The blinder generates a blinded public key, asks the signer to generate a signature, which he can transform into a signature for the blinded public key by applying the isometry he used for blinding. To verify the signature, the blinder has to provide the proof, that the blinded public key is related to the original public key and that he did the blinding according to the specification of the algorithm. The latter can be done by adding the blinded public key and a static zero knowledge proof for the isometric subcode connection to the signature.

Our blind signature conversion has two major drawbacks: The large signatures (as we have to provide static zero knowledge proofs) and the slow blind signature generation: In order to turn the unblinding impossible for the signer, we have to assure that each signature produced by the signer can be transformed into any blind signature by an appropriate isometry. This condition is fulfilled only if all signature vectors generated by the signer are of the same norm and each vector of a certain norm can be transformed in each vector of the same norm. The latter is only possible in Hamming or Euclidean norm, which limits the choice of signature schemes to which we may apply our blind signature conversion.

Related work Related work, as mentioned above are generic constructions of blind signature schemes. The only ones, we are aware of are

– Weak blind signatures, which involve trusted third parties [3] and

– Blind signatures from PKCs [5], which apply only to PKCs which are trapdoor permutations. All known PKCs satisfying this condition are based on number theory.

Organization The paper is organized as follows: In the next sections we present our notation for codes, define some $\mathcal{NP}$-hard problems and recall the basic schemes we use as a tool-box. In the third section we present our conversion and apply our conversion to the CFS scheme in Section 4. At the end we draw conclusions, make some remarks on how to extend the conversion to NTRUSign and give some hints on possible next steps in research.

2 Preliminaries

This section will introduce the basic notation, schemes and hard problems we use throughout the paper. With a slight abuse of notation we will not make a difference between a code and its generator matrix. We shortly define our notation for codes:

Definition 2.1. *Let $\mathcal{C}$ be an $[n,k]$ code over $\mathbb{F}_q$, i.e. a k-dimensional subvectorspace of $\mathbb{F}_q^n$. A $k \times n$ matrix C is a generator matrix of $\mathcal{C}$ if the rows of C span $\mathcal{C}$. We denote with $\mathsf{C}^{\perp}$ the check matrix of $\mathcal{C}$, i.e. the $n \times (n-k)$ matrix in systematic form, such that $\mathsf{C} \cdot \mathsf{C}^{\perp}$ is the zero matrix.*

In the following the norm used will be the Hamming norm and denoted by $\|\cdot\|$, but one could think of different norms as well.

2.1 Underlying problems

In this section we present the problems on which the security of our blind signature conversion will be based. The hardness of all these problems can be reduced to the hardness of *General Syndrome Decoding*, which is $\mathcal{NP}$-hard even in the binary case [6]. Here we give a definition polynomial-time equivalent to the standard one:

Definition 2.2. *Let G be the generator matrix of an $[n,k]$ code over a finite field $\mathbb{F}$ with check matrix H, $w \in \mathbb{N}$ and $\mathbf{s}$ be an $n-k$ vector over $\mathbb{F}$ called syndrome. The Problem of General Syndrome Decoding (GSD) is to find a vector $\mathbf{e} \in \mathbb{F}^n$ of Hamming weight $\leq w$ such that the syndrome of $\mathbf{e}$ is $\mathbf{s}$, i.e. $\mathsf{H}(\mathbf{e}) := \mathbf{e}\mathsf{H} = \mathbf{s}$.*

To obtain a blind signature scheme, we will have to rely on the hardness of another problem, the *Permuted Kernel Problem*:

Definition 2.3. *The Permuted Kernel Problem (PKP) is defined as follows: Let* $\mathsf{A}^{\perp} \in \mathbb{Z}_q^{(n-k)\times n}$ *be a matrix and* $\mathbf{v} \in \mathbb{Z}_q^n$ *a vector, where* $k < n, q \in \mathbb{N}$*. Find a permutation matrix* P*, such that* $\mathbf{v}\mathsf{P}\mathsf{A}^{\perp} = \mathbf{0}$*, i.e.* $\mathbf{v}\mathsf{P}$ *is in the vector space spanned by the rows of* $\mathsf{A} = (\mathsf{A}^{\perp})^{\perp}$*. We write* $\mathsf{A} > \mathbf{v}\mathsf{P}$ *or* $\mathsf{A} >^{\mathsf{P}} \mathbf{v}$*.*

The PKP is $\mathcal{NP}$-hard even if $q = 2$ (since a polynomial time solver for the binary PKP can solve the $\mathcal{NP}$-hard binary general syndrome decoding problem in polynomial time) or if $k = n - 1$ [7]. It is easy to verify that a decisional variant of the PKP is polynomial time equivalent to PKP – even in the case of signed permutations.

It is nice to remark, that NTRUEncrypt and NTRUSign can be broken if the PKP can be solved efficiently as the distribution of the entries in the secret NTRU vector is a system parameter. To apply our conversion to NTRUSign, we would need analogeous definitions for genereral isometries, not just permutations. We will come back to this issue in the conclusion. For now, we would like to remark that the criteria for an isometry change depending on the norm used (for any $\mathcal{L}_p$ norm with $l \neq 2$, all isometries are signed permutations).

2.2 Digital signatures and blind signatures

We now define the cryptographic primitives employed in our paper with the exception of a zero knowledge proof system, where we refer to [4].

Definition 2.4. *A signature scheme based on codes consists of three algorithms:*

- *A key generation algorithm which on input of the security parameter(s) returns a private key and a public key consisting of a check matrix* $\mathsf{C}_{\mathrm{pub}}^{\perp}$ *and an integer* B*.*
- *A signature algorithm, that on input of a (hash of a) message* $\mathbf{m}$ *and the private key returns a vector* $\mathbf{y}$ *of norm at most* B *satisfying* $\mathbf{m} = \mathsf{C}_{\mathrm{pub}}^{\perp}(\mathbf{y})$*.*
- *A verification algorithm, which on input of of* $\mathbf{m}$ *and* $\mathbf{y}$ *returns accept iff* $\mathbf{m} = \mathsf{C}_{\mathrm{pub}}^{\perp}(\mathbf{y})$ *and* $\|\mathbf{y}\| \leq B$*, reject otherwise.*

We consider a signature scheme to be secure if its signatures

(i) are *authentic*, i.e. they convince that the signer interacted in generating them,

(ii) are *non-malleable* and *unforgeable* in the sense, that no one but the signer can generate a $l+1$st signature knowing l signatures and
(iii) can not be *repudiated*, i.e. the signer should not be able to claim that he did not generate them.

The CFS digital signature meets the conditions of a secure code based signature scheme according to the above definition. We expect the reader to be familiar with that scheme which we recall briefly at the end of the section.

In our paper however, we consider a slightly different flavor of a blind signature scheme:

Definition 2.5. *A blind signature scheme consists of two parties, namely the blinder and the signer, and six algorithms:*

- *A key generation algorithm, that on input of the security parameters returns a public/private key pair, hands the private key to the signer and publishes the public key.*
- *A blinding algorithm, in which the blinder on input of the public key and (the hash of) a message* $\mathbf{m}$ *returns a blinded message* $\mathbf{m}^{\text{blind}}$ *and a unblinding information* $(\mathbf{m}, \mathbf{u})$.
- *A signing algorithm, that on input of the private key and* $\mathbf{m}^{\text{blind}}$ *returns a signature* $\mathbf{y}$.
- *A verification algorithm, that on input of the public key, a (hash of a) message* $\mathbf{m}'$ *and a signature* $\mathbf{y}$ *returns accept or reject*
- *An unblinding algorithm, where the blinder on input of* $\mathbf{y}$, $(\mathbf{m}, \mathbf{u})$ *and the public key returns a blind signature* σ *if* $\mathbf{y}$ *is verifiable by the above algorithm with the public key and* $\mathbf{m}^{\text{blind}}$.
- *A blind verification algorithm, that on input of a blinded signature* σ, *a (hash of a) message* $\mathbf{m}'$ *and the public key returns accept or reject.*

We say that a blind signature scheme works correct if every blind signature σ *generated by the unblinding algorithm is verifiable with the blind verification algorithm and the corresponding message.*

Oftentimes there is no difference between the verification and the blind verification algorithm (like e.g. in the blind signature scheme presented in [1]) – however there is no need that the two algorithms are identical.

The classical but maybe somewhat out of date definition of a secure blind signature scheme (by Chaum) is the following:

Definition 2.6. *A blind signature scheme is secure if the following three requirements are met:*

(i) Digital signature – If the blind verification algorithm returns accept, anyone can check, that σ was formed using the signers private key.

(ii) Blind signature – The signer can not learn anything about the correspondence between the signatures $\mathbf{y}_i$ and the blind signatures σ_i without the unblinding information.

(iii) Conservation of signature – The blinder can create at most (!) one blind signature for each signature returned by the signer.

2.3 Basic protocols

In the following we will employ **Shamir's Permuted Kernel ZKIP** [7], whose basic variant works as follows: Given the $[n, k]$ code G and an n-vector $\mathbf{v}$ over $\mathbb{F}_q$ Shamir's protocol can be used to proof in zero knowledge, that there is a permutation $\mathsf{P} \in \mathcal{S}_n$, such that $\mathbf{v}\mathsf{P}$ is in G. By Shamir's 5-round zero-knowledge protocol, on can prove the knowledge of P using two blending factors: a permutation Π and a random n-vector over $\mathbb{F}_q$. However, a dishonest prover not knowing P can cheat the verifier in the protocol with probability $(q+1)/(2 \cdot q)$. Thus, the protocol has to be repeated several times to detect cheating provers. Computing Π from $\mathbf{v}$ and G is is solving the PKP. The communication cost is about $n(l + \log_2(n)) \log_2(q)$ plus two times the size of the commitments.

Since we can represent a $[n, l]$ subcode of G as an n-vector over $\mathbb{F}_{q^l}$ and G over $\mathbb{F}_{q^l}$ as well, we can prove the permuted subcode connection by Shamir's protocol. In each iteration, a cheating prover has the probability of $(q^l+1)/(2 \cdot q^l)$ to succeed. In the following q will be 2, such that the terms "premutation" and "isometry" coincide for the Hamming norm. However, if $q \neq 2$, Shamir's protocol works analogeous for general isometries in Hamming norm.

In Section 4 we will apply our conversion to the **CFS signature** [2], which we explain in short. It starts from a secret Goppa code with generator matrix $\mathsf{G} \in \mathbb{F}_2^{k \times n}$ and known error correction algorithm. For an error $\mathbf{e} \in \mathbb{F}_2^n$ we define its *syndrome* as $\mathbf{s} = \mathbf{e}\mathsf{G}^{\perp}$. If $\mathbf{e}$ is of weight below the error correcting radius t, it is uniquely defined by its syndrome and can efficiently recovered from $\mathbf{s}$ by the error correction algorithm of G. The public key is the systematic check matrix H^{pub} of the permuted code GP with a secret permutation $\mathsf{P} \in \mathcal{S}_n$. To sign a syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$, the signer tries to compute a vector $\mathbf{e}$ of weight $\leq B$ with syndrome $\mathbf{e}\mathsf{H}^{\text{pub}} = \mathbf{s}$. Otherwise the signature generation fails.

In practice, since not for all syndromes $\mathbf{s}$ there is an error vector $\mathbf{e}$ of weight $\leq B$, which can be computed efficiently, a random vector $\mathbf{r}$ is chosen and the signer tries to sign $\mathbf{s}' = h(\mathbf{m}||\mathbf{r})$ instead, where h is a hash function. The choice of $\mathbf{r}$ has to be repeated until a signature can be generated, which is about $(\binom{2^m}{B} \cdot 2^{-mt})^{-1} \approx t!$ times if $B = t$. The resulting signature is then $(\mathbf{r}, \mathbf{e})$.

Besides the slow signature generation, typical parameter sets for CFS like $n = 2^{16}, k = n - 16 \cdot 9, t = 9$ are subject to attacks via Wagner's solution of the generalized birthday problem as stated by Bleichenbacher. Consequently, an attack on this parameter set can be performed in about $2^{59.6}$ operations.

3 The blind signature conversion

In this section we show how to blind a signature scheme based on solving GSD. We will take such a signature scheme as a black box and convert it by adding a blinding, unblinding and blind verification algorithm.

The general idea is as follows: Given the public generator matrix C_{pub} of the signature scheme, we add a number of vectors R_{pub}, which are generated by a pseudo random number generator. While the signer still can solve the GSD in the resulting code, we can at the same time blind this code by taking a permuted (isometric) subcode $\mathsf{C}_{\text{blind}}$. Knowing the employed permutation $\mathsf{\Pi}$, the signer still would be able to solve the GSD in $\mathsf{C}_{\text{blind}}$ as long as C_{pub} is a isometric subcode of $\mathsf{C}_{\text{blind}}$:

$$\mathsf{C}_{\mathsf{R}} := \left[\frac{\mathsf{C}_{\text{pub}}}{\mathsf{R}_{\text{pub}}}\right] > \mathsf{C}_{\text{blind}}\mathsf{\Pi}^{-1} > \mathsf{C}_{\text{pub}}.$$

To generate a blind signature of a syndrome $\mathbf{s}$, the blinder can thus do the following to solve the GSD in $\mathsf{C}_{\text{blind}}$ in interaction with the signer: Take the syndrome $\mathbf{s}$ of $\mathsf{C}_{\text{blind}}$ and convert it into a syndrome of C_{pub}, which can be signed by the signer. Then, apply the isometry $\mathsf{\Pi}$ to the signature $\mathbf{y}$ to obtain the GSD solution $\mathbf{y}\mathsf{\Pi} \in \mathsf{C}_{\text{blind}}$ for the challenge $\mathbf{s}$. We will denote with PKP-Proof $(\mathsf{C}^{\perp}_{\text{blind}}, \mathsf{C}^{\perp}_{\mathsf{R}})$ the static PKP-Proof for $\mathsf{C}_{\text{blind}}$ being an isometric subcode of C_{R}.

We present our conversion of a signature scheme based on codes lattices to a blind signature scheme in Algorithms 3.1, 3.2 and 3.3. After the blinding algorithm, $\mathbf{s}$ is sent to the signer who returns the signature $\mathbf{y}$ of norm B for $\mathbf{s}$, which is verifiable with $\mathsf{C}^{\perp}_{\text{pub}}$: $\mathsf{C}^{\perp}_{\text{pub}}(\mathbf{y}) = \mathbf{s}$ (We assume the

Algorithm 3.1 Blinding algorithm

System parameters: $l < L$, Hash function h, a pseudo random number generator and a norm $\|\cdot\|$.
Input: The message $\mathbf{m}$, the public matrix $\mathsf{C}^{\perp}_{\text{pub}}$, $\mathbf{r}$ a random seed and B.
Output: The blinded syndrome $\mathbf{s}$ and the unblinding information $\mathbf{u}$.

- Generate the matrix $\mathsf{R}_{\text{pub}} \in \mathbb{Z}_q^{L \times n}$ from $\mathbf{r}$ by a pseudo random number generator.
- Generate a random $l \times L$ full rank matrix S over $\mathbb{Z}_q$, set $\mathsf{R} = \mathsf{S}\mathsf{R}_{\text{pub}}$.
- Generate a random isometry $\mathsf{\Pi}$ of the norm $\|\cdot\|$ (i.e. a permutation in the case of the Hamming norm).
- Generate the check matrix $\mathsf{C}^{\perp}_{\text{blind}}$ of the code with generator matrix

$$\mathsf{C}_{\text{blind}} = \left[\frac{\mathsf{C}_{\text{pub}}}{\mathsf{R}}\right] \mathsf{\Pi}.$$

- Compute the blinded syndrome $\mathbf{s}$, such that for all $\mathbf{z} \in \mathbb{Z}^n$

$$\left(\mathsf{C}^{\perp}_{\text{pub}}(\mathbf{z}) = \mathbf{s}\right) \Rightarrow \left(\mathsf{C}^{\perp}_{\text{blind}}(\mathbf{z}\mathsf{\Pi}) = h(\mathbf{m}||\mathsf{C}^{\perp}_{\text{blind}})\right)$$

via linear algebra:
- solve $\mathsf{C}^{\perp}_{\text{blind}}(\mathbf{z}) = h(\mathbf{m}||\mathsf{C}^{\perp}_{\text{blind}})$ for some $\mathbf{z}$.
- set $\mathbf{s} = \mathsf{C}^{\perp}_{\text{pub}}(\mathbf{z}\mathsf{\Pi}^{-1})$.

return the blinded syndrome $\mathbf{s}$ and the unblinding information $\mathbf{u} = (\mathbf{r}, \mathsf{\Pi}, \mathsf{C}^{\perp}_{\text{blind}})$

Algorithm 3.2 Unblinding algorithm

System parameters: $l < L$, Hash function h, a pseudo random number generator and a norm $\|\cdot\|$.
Input: The message $\mathbf{m}$, the public matrix $\mathsf{C}^{\perp}_{\text{pub}}$, B, the blinded syndrome $\mathbf{s}$, the unblinding information $\mathbf{u} = (\mathbf{r}, \mathsf{\Pi}, \mathsf{C}^{\perp}_{\text{blind}})$ and a valid signature $\mathbf{y}$ of $\mathsf{C}^{\perp}_{\text{pub}}$.
Output: The blind signature σ of $\mathbf{m}$ or `failure`.

- veritfy that $\mathbf{y}$ has norm B and $\mathsf{C}^{\perp}_{\text{pub}}(\mathbf{y}) = \mathbf{s}$ else **return** `failure`
- Generate PKP-Proof $(\mathsf{C}^{\perp}_{\text{blind}}, \mathsf{C}^{\perp}_{\mathsf{R}})$ and **return** the blind signature

$$\sigma = \left(\mathbf{r},\, \mathsf{C}^{\perp}_{\text{blind}},\, \bar{\boldsymbol{\sigma}} = \mathbf{y}\mathsf{\Pi},\, \text{PKP-Proof}\ (\mathsf{C}^{\perp}_{\text{blind}}, \mathsf{C}^{\perp}_{\mathsf{R}})\right).$$

signer to halt the protocol if $\mathbf{y}$ does not have norm B of if he is unable to compute it).

Please observe, that with a fixed $\mathbf{r}$, there are q^l possible $\mathbf{s}$ which result from the possibilities to add a vector from the space spanned by R to $\mathbf{z}$. However, it is not desirable to use the same R (and thus $\mathsf{C}_{\text{blind}}$) twice (e.g. if the signing algorithm fails) since the signer could identify R after receiving l syndromes built with the same R. Further, if the signature σ

Algorithm 3.3 Blind Verification

Input: L, the hash function h, a norm $\|\cdot\|$, $\mathbf{m}$, $\mathsf{C}^{\perp}_{\text{pub}}$ and the blind signature $\sigma = \left(\mathbf{r},\, \mathsf{C}^{\perp}_{\text{blind}},\, \bar{\boldsymbol{\sigma}},\, \text{PKP-Proof}\,(\mathsf{C}^{\perp}_{\text{blind}}, \mathsf{C}^{\perp}_{\mathsf{R}})\right)$

- Generate the matrix $\mathsf{R}_{\text{pub}} \in \mathbb{Z}_q^{L\times n}$ from $\mathbf{r}$ by a pseudo random number generator.
- Generate some vector $\bar{\mathbf{y}}$ such that $\mathsf{C}^{\perp}_{\text{blind}}(\bar{\mathbf{y}}) = h(m \| \mathsf{C}^{\perp}_{\text{blind}})$.
- Check $\|\bar{\boldsymbol{\sigma}}\| < A$ and $\bar{\mathbf{y}} - \bar{\boldsymbol{\sigma}} \in \mathsf{C}_{\text{blind}}$.
- Check PKP-Proof $(\mathsf{C}^{\perp}_{\text{blind}}, \mathsf{C}^{\perp}_{\mathsf{R}})$

was generated according to algorithm 3.1, the verification in Algorithm 3.3 does not fail in the case of an honest blinder. Our scheme thus satisfies the condition of being a signature. We will prove the security of that signature in the following.

3.1 Security analysis

The signature conversion can be proven secure to a certain extend if the underlying signature scheme is secure, as we will show in this section. We first state the security argument for the signer:

Theorem 3.1. (Conservation of signature) *Let $\sigma_i, i = 1, \cdots, m$ be a set of blind signatures and $\mathbf{y}_j, j = 1, \cdots, m$ be a set of signatures used to generate the σ_i. If the blinder can produce a $m+1$st blind signature σ_{m+1} without interaction with the signer, then he can solve the GSD for C_{R}, B and a random vector $\mathbf{v}$.*

The theorem above is easy to verify thus we omit giving a formal proof. As long as GSD with input C_{R} and $w = B$ is infeasible to solve without knowing the secret key for C_{pub}, the blinder can not deduce more blind signatures, than the ones he created in interaction with the signer. The security argument for the blinder is much harder to prove:

Theorem 3.2. (Blind Signature) *Assume that it is infeasible to prove that the minimum distance of C_{pub} is below B. Let $\sigma_i, i = 1, \cdots, m$ be a*

set of blind signatures and $\mathbf{y}_j, j = 1, \cdots, m$ *be a set of signatures used to generate the* σ_i*. If the signer can identify the* m *pairs* (i, j)*, such that* $\mathbf{y}_j$ *was used to generate* σ_i*, then he can solve at least one of the (decisional) PKP Problems:*

$$\left(\mathsf{C}_{\sigma_i} = \begin{bmatrix} (\mathsf{C}_{\text{blind}})_i \\ \mathbf{y}_i \mathsf{\Pi}_i \end{bmatrix}, \mathbf{y}_j\right)$$

Proof. We know that the zero-knowledge PKP proofs do not reveal any information on the correct pairs (i, j). Thus, if the signer can identify the correct pairs, he is able to solve the PKP instances

$$\begin{bmatrix} \mathsf{C}_{\text{pub}} \\ \mathsf{R}_{\text{pub}} \\ \mathbf{y} \end{bmatrix} >^{\mathsf{\Pi}} \begin{bmatrix} \mathsf{C}_{\text{blind}} \\ \mathbf{y}\mathsf{\Pi} \end{bmatrix} >^{\mathsf{\Pi}^{-1}} \begin{bmatrix} \mathsf{C}_{\text{pub}} \\ \mathbf{y} \end{bmatrix} >^{\mathsf{Id}} \mathbf{y} \text{ or } \quad \begin{bmatrix} \mathsf{C}_{\text{pub}} \\ \mathsf{R}_{\text{pub}} \end{bmatrix} >^{\mathsf{\Pi}} \mathsf{C}_{\text{blind}} >^{\mathsf{\Pi}^{-1}} \mathsf{C}_{\text{pub}}.$$

However, solving the first sequence of permuted inclusions implies solving the second one: Assume, the efficient algorithm solving the first sequence gives an answer $\hat{\mathsf{\Pi}}$ such that

$$\begin{bmatrix} \mathsf{C}_{\text{blind}} \\ \mathbf{y}\mathsf{\Pi} \end{bmatrix} > \begin{bmatrix} \mathsf{C}_{\text{pub}} \\ \mathbf{y} \end{bmatrix} \hat{\mathsf{\Pi}} \text{ and } \quad \mathsf{C}_{\text{blind}} \not> \mathsf{C}_{\text{pub}} \hat{\mathsf{\Pi}}.$$

Then, $\mathbf{y}\mathsf{\Pi}\hat{\mathsf{\Pi}}^{-1}$ is in C_{pub}. This however means that we have found a vector of norm B in C_{pub}, which is below the known minimum distance of C_{pub} and would thus contradict the statement of the theorem.

4 Application to the CFS signature scheme – an example

In this section we will briefly view the application of our conversion to the CFS signature scheme. As example we start from the parameter set $m = 16, t = 9$ and $B = 9$. The public key $\mathsf{C}^{\perp}_{\text{pub}}$ is a $2^m \times mt$ binary check matrix, which takes about 1.12 Mbytes to be stored. If we take, for example $2l = L = 80$ as parameter set for our blind signature conversion, then $\mathsf{C}^{\perp}_{\text{blind}}$ will be a $2^{16} \times 104$ binary matrix, which takes about 0.81 Mbytes to be stored. We have $d = 2t+1 > B$, thus the conditions of Theorem 3.2 are met. Algorithm 3.1 has to be called about $(\binom{2^m}{B} 2^{-mt})^{-1} \approx (t!)^{-1} \approx 2^{18.5}$ times to generate a single signature, where each call costs the signer $m^3 t^2 \approx 2^{18.33}$ and the blinder $(16 \cdot 9)^3 \approx 2^{21}$ operations, which is the dominating part of the blind signature generation (The blinder has to change only a few rows of $\mathsf{C}_{\text{blind}}$ per call, not all – the only condition is that he does not use the same $\mathsf{C}_{\text{blind}}$ twice). Thus, a signature can be generated in time corresponding to eight CFS signatures. The size of a signature is

dominated by PKP-Proof $(\mathsf{C}_{\mathrm{blind}}^{\perp}, \mathsf{C}_{\mathsf{R}}^{\perp})$, which requires to store a matrix of the size of $\mathsf{C}_{\mathrm{blind}}$ each round. The resulting signature has thus a size of about 50 Mbytes. The scheme reaches more than 2^{80} security against unblinding, as so far, there is no effective attack to solve the PKP problem for codes. The fastest attack is by the generalized birthday paradox, aims to forge blind signatures and requires $2^{59.6}$ operations like for the original CFS scheme – remember that the blind signature generation takes $\approx 2^{39.5}$ Operations.

Changing t to 10 and B to 12 would reduce the number of calls of Algorithm 3.1 before a valid blind signature is generated to $(\binom{2^m}{B} \cdot 2^{-mt})^{-1} \approx 1/8 \leq 1$. At the same time, this raises the signer's workfactor (he has to guess two error positions and correct another 10 ones do decode a syndrome with $B = 12$ errors). Now, the signature generation takes only about twice as many operations as before $(\binom{B}{B-t}/\binom{2^m}{B-t} \cdot m^3 t^2 \approx 2^{40.4})$, while the security against existential forgery increases to about $2^{73.6}$.

Even if the security level of the proposed blind signature with the CFS scheme is sufficiently high, blind signature generation and storage is too costly to consider our scheme for practical applications.

5 Conclusion

In this paper we present the first conversion of a non-number theoretic signature scheme to a blind signature scheme without a TTP. Our conversion is applicable to code based signature schemes. However, there are two drawbacks of our solution: The slow blind signature generation (as we require all signatures to be of the same norm) and the large signature sizes (as the signatures include a static PKP-Proof). Of course, the security of the proposed blind signature conversion against quantum computers is not proven but only conjectured and thus further research has to be done.

Possible extensions to lattice based signature schemes As we did already remark, the concept of our conversion to a blind signature scheme could be extended to lattice based signature schemes like NTRUSign. Essential for the application to other schemes than CFS is the possibility to transform two vectors of the same norm into each other via isometries. This is only possible for special norms like the Hamming and Eucledian norm.

The crucial point is, that the PKP protocol can be extended to isometries. Further, the concept of syndromes can be transfered to lattices since the scalar product of some vector with any vector of the dual lattice is an integer iff the vector is in the lattice. The major obstacle for applying such an conversion e.g. to NTRUSign is the fact, that NTRUSign produces signature vectors of variable length. Thus, one would have to consider a variant of NTRUSign which produces signature vectors of a single length, only. So far we can not give bounds on the fraction of NTRU signatures of a certain norm better than the obvious lower bound $1/B^2$ (the square of the signature vector's norm has to be an integer), where B refers to the maximal norm of accepted signature vectors.

Further research and open questions One could improve significantly on the problem of the signature sizes, if one could provide a zero knowledge proof for the PKP problem, which requires less rounds, i.e. with lower cheating probability. This could be possible due to the fact, that the kernel we use is of dimension > 1.

Another direction of research could be to speed-up the signature generation by allowing signatures of different norms, e.g. by allowing the blinder to add a short blending vector to the signature. However, so far we were not able to give evidence for the security of such a variant.

References

1. David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.
2. N. Courtois, M. Finiasz, and N.Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248, pages 157–174. Springer-Verlag, 2001.
3. Matthew K. Franklin and Moti Yung. The blinding of weak signatures (extended abstract). In *EUROCRYPT 1994*, pages 67–76, 1994.
4. O. Goldreich. *Modern Cryptography, Probabilistic Proofs an Pseudo-randomness.* Springer-Verlag Heidelberg, 1999.
5. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1997.
6. F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correctiong Codes.* North-Holland Amsterdam, 7 edition, 1992.
7. A. Shamir. An efficient identification scheme based on permuted kernels. In *Proc. of Crypto'89*, volume 435 of *LNCS*, pages 606–609. Springer Verlag, 1990.