Efficient Post-quantum Blind Signatures

Abstract. We present the first efficient post-quantum blind signature scheme. Our scheme is provably secure in the random oracle model, unconditionally blind, and round-optimal. We propose it as a replacement for current blind signature schemes for the post-quantum era. Its basis of security is a problem related to finding short vectors in a lattice.

Keywords. Post-quantum cryptography, blind signatures, lattices

1 Introduction

Since 1982, when Chaum proposed his idea of blind signatures [Cha83], it has become an important primitive for anonymous Internet banking and e-voting applications. These applications will retain their importance in both, near and far future. As for the near future, we are convinced that current factoring and discrete logarithm based instantiations are efficient and secure. *But for how long?* Today, when building provably secure cryptographic schemes, one has to keep emerging technologies and especially quantum computers in mind. In the quantum-age, the cryptographic assumptions change with the leap in computing power that quantum computers will provide.

There are only a few cryptographic assumptions that are conjectured to be *post-quantum*, i.e. they are considered to be able to withstand quantum computer attacks. One of those assumptions is the hardness of finding short vectors in a lattice.

Our contribution and related work. Using a problem that is related — yet not equivalent — to the shortest vector problem (SVP) in a lattice as our security assumption, we construct the first post-quantum blind signature scheme. According to the security model, mainly influenced by Juels, Luby, and Ostrovsky [JLO97] as well as Pointcheval and Stern [PS00], blind signature schemes have to satisfy blindness and one-more unforgeability. Blindness states that the signer must not obtain any information on the signed messages and one-more unforgeability enforces that an adversarial user cannot obtain more signatures than there were interactions with the signer.

Our scheme is unconditionally blind, i.e. even secure against computationally unbounded adversaries. In order to prove unforgeability, we use a computational problem similar to the "one-more" problems [BNPS03] in the RSA context. As for its efficiency, we state that it is as efficient as the underlying signature scheme proposed by Gentry, Peikert, and Vaikuntanathan (GPV) [GPV08] and with its two rounds, it is even *round-optimal*. The security of the GPV signature scheme is proven in the random oracle model and, due to Ajtai's result [Ajt96], is based on the worst-case hardness of the SVP. Ajtai showed that solving the average-case SVP is at least as hard as solving a related problem in the worst-case in lattices of a certain smaller dimension — a unique and desirable

property of lattice cryptography. The works of Micciancio and Regev [MR07] and [GPV08] improve the tightness of this reduction. Assuming, e.g., that *worst-case* instances of standard lattice problems in dimension $n = 64$ are hard (c.f. [LMPR08]), the scheme's private and public key need about 6 kilobytes of storage and the signatures size is around 13 kilobytes. Signature computation would then, roughly speaking, involve two matrix-vector multiplications modulo 262,147, involving a 1733×64 matrix, which can be done in about 500,000 integer operations. Thus, signing and verifying should take less than 1 ms on a modern CPU[1]. When SIMD[2] instructions are available, this can be significantly reduced. Verification is even more efficient.

We believe that our work is an important contribution and that we solve a longstanding problem because the previous efficient constructions, like [Cha83, PS97, PS00, Abe01, BNPS03, CKW04, KZ05, Oka06], have one thing in common: they are built upon classic number theoretic assumptions, like the hardness of factoring large integers or computing discrete logarithms. The newer approaches of Boldyreva [Bol03] and Okamoto [Oka06] tend to use pairings and bilinear maps that yield very elegant constructions. They, however, are again based on the discrete logarithm problem in this specific setting. None of the above schemes remain secure in the presence of reasonably large quantum computers, where both factoring and computing discrete logarithms becomes easy due to the seminal work of Shor [Sho97].

Finally, we would like to mention that there are also (inefficient) instantiations from general assumptions, e.g. by Juels, Luby, and Ostrovsky [JLO97], Fischlin [Fis06], and Hazay, Katz, Koo, and Lindell [HKKL07]. Whether they are post-quantum, largely depends on the exact realization of primitives.

Organization. After a preliminaries section, we present our construction in Section 3. There, we also prove that our scheme has the well-established security properties. In Section 4, we discuss the security of the underlying signature scheme under our modifications.

2 Preliminaries

With n, we always denote the security parameter. $(a, b) \leftarrow \langle \mathcal{A}(x), \mathcal{B}(y) \rangle$ denotes the joint execution of two algorithms $\mathcal{A}$ and $\mathcal{B}$ in an interactive protocol with private inputs x to $\mathcal{A}$ and y to $\mathcal{B}$. The private outputs are a for $\mathcal{A}$ and b for $\mathcal{B}$.

In the following, we recall the definitions of digital signature schemes and of blind signature schemes along with their respective security models. A brief introduction to lattice theory is given in Section 4.

2.1 Digital signatures

A digital signature scheme DS is a triple $(\mathsf{Kg}, \mathsf{Sig}, \mathsf{Vf})$ where

Key generation. $\mathsf{Kg}(n)$ outputs a private signing key sk and a public verification key pk.

Signature issue. $\mathsf{Sig}(\mathsf{sk}, M)$ outputs a signature σ on a message M from the message space $\mathcal{M}$ under sk.

[1] Running at 2.4 GHz, doing one integer multiplication in 4 cycles.
[2] Single Instruction, Multiple Data – allows faster scalar products.

Signature verification. The algorithm $\mathsf{Vf}(\mathsf{pk}, \sigma, M)$ outputs 1 if σ is a valid signature on M under pk and otherwise 0.

Signature schemes are complete, i.e. for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kg}(n)$, all messages $M \in \mathcal{M}$, and any $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, M)$, we have $\mathsf{Vf}(\mathsf{pk}, \sigma, M) = 1$.

Security of digital signature schemes is typically proven against existential forgery under a chosen message attack (EU-CMA), where an adversary wins if it outputs a signature on a *new* message M^* after accessing a signature oracle on a polynomial number of different messages. For the underlying signature scheme, we need the notion of strong unforgeability under a chosen message attack (SU-CMA), where the adversary even wins if it is able to output a *new pair* (M^*, σ^*), i.e. it is not forced to output a signature on a new message. In the random oracle model, the adversary has access to a hash oracle H that is chosen from the family of all collision resistant hash functions $\mathcal{H}(n)$. This is formalized in the following experiment.

Experiment $\mathsf{Exp}^{\mathsf{su\text{-}cma}}_{\mathcal{A},\mathsf{DS}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(n)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kg}(n)$
$(M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{H}(\cdot), \mathsf{Sig}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
let (M_i, σ_i) be the answer returned by $\mathsf{Sig}(\mathsf{sk}, \cdot)$ on input M_i, for $i = 1, \ldots, k$.
Return 1 iff $\mathsf{Vf}(\mathsf{pk}, M^*, \sigma^*) = 1$ and $(M^*, \sigma^*) \notin \{(M_1, \sigma_1), \ldots, (M_k, \sigma_k)\}$.

The scheme DS is called $(t, q_{\mathsf{Sig}}, q_{\mathsf{H}}, \epsilon)$-strongly unforgeable if there is no adversary, running in time at most t while making at most q_{Sig} queries to the oracle $\mathsf{Sig}(\mathsf{sk}, \cdot)$ and at most q_{H} queries to the hash oracle, that succeeds in the above experiment with probability at least ϵ.

2.2 Blind signatures

A blind signature scheme BS consists of three algorithms $(\mathsf{Kg}, \mathsf{Sig}, \mathsf{Vf})$, where Sig is an interactive protocol between a signer $\mathcal{S}$ and a user $\mathcal{U}$. The specification is as follows.

Key generation. $\mathsf{Kg}(n)$ outputs a private signing key sk and a public verification key pk.

Signature issse. $\mathsf{Sig}(\mathsf{sk}, M)$ describes the joint execution of $\mathcal{S}$ and $\mathcal{U}$. The private output of $\mathcal{S}$ is a view $\mathcal{V}$ and the private output of $\mathcal{U}$ is a signature σ on the message M under sk. Thus, we write $(\mathcal{V}, \sigma) \leftarrow \langle \mathcal{S}(\mathsf{sk}), \mathcal{U}(\mathsf{pk}, M) \rangle$.

Signature verification. Same as in DS.

Completeness is defined as with digital signature schemes. Views are interpreted as random variables, whose output is generated by subsequent executions of the respective protocol. Two views $\mathcal{V}_1$ and $\mathcal{V}_2$ are considered equal if they cannot be distinguished by any computationally unbounded algorithm with noticeable probability.

As for security, blind signatures have to satisfy two properties: blindness and one-more unforgeability [JLO97, PS00]. The notion of blindness is defined in the following experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}$, where the adversarial signer $\mathcal{S}^*$ chooses two messages M_0, M_1 and interacts with two users who obtain blind signatures for the two messages in random order. Note that the executions of the two users may be arbitrarily interleaved. After seeing the unblinded signatures in the original order, with respect to M_0, M_1, the signer has to guess the message that has been signed for the first user.

Experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}(n)$
$b \leftarrow \{0,1\}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{BS.Kg}(n)$
$(M_0, M_1) \leftarrow \mathcal{S}^*(\mathsf{pk},\mathsf{sk})$
$(\mathcal{V}_0, \sigma_b) \leftarrow \langle \mathcal{S}^*(\mathsf{sk}), \mathcal{U}_0(\mathsf{pk}, m_b)\rangle$
$(\mathcal{V}_1, \sigma_{1-b}) \leftarrow \langle \mathcal{S}^*(\mathsf{sk}), \mathcal{U}_1(\mathsf{pk}, m_{1-b})\rangle$
Either of the signatures might equal fail.
If $\sigma_0 \neq \mathsf{fail}$ and $\sigma_1 \neq \mathsf{fail}$
$d \leftarrow \mathcal{S}^*(\mathsf{sk},\mathsf{pk},\sigma_0,\sigma_1)$
Else
$d \leftarrow \mathcal{S}^*(\mathsf{sk},\mathsf{pk},\mathsf{fail},\mathsf{fail})$
Return 1 iff $d = b$

A signature scheme BS is (t,ϵ)-blind, if there is no adversary $\mathcal{S}^*$, running in time at most t, that wins the above experiment with advantage at least ϵ, where the advantage is defined as

$$\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}} = \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}(n) = 1] - \frac{1}{2}\,.$$

The second security property, one-more unforgeability, ensures that each completed interaction between signer and user yields at most one signature. It is formalized in the following experiment $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}$, where an adversarial user tries to output $\jmath$ valid signatures after $\ell < \jmath$ completed interactions with an honest signer.

Experiment $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(n)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{BS.Kg}(n)$
$\{(M_1,\sigma_1),\ldots,(M_\jmath,\sigma_\jmath)\} \leftarrow \mathcal{U}^{*\mathsf{H}(\cdot),\langle\mathcal{S}(\mathsf{sk}),\cdot\rangle}(\mathsf{pk})$
Let ℓ be the number of (complete) interaction between $\mathcal{U}^*$ and the signer.
Return 1 iff
1. $m_i \neq m_j$ for all $1 \le i < j \le \jmath$;
2. $\mathsf{BS.Vf}(\mathsf{pk},\sigma_i,M_i) = 1$ for all $i = 1,\ldots,\jmath$;
3. $\ell < \jmath$.

A signature scheme BS is $(t, q_{\mathsf{Sig}}, q_{\mathsf{H}}, \epsilon)$-one-more unforgeable if there is no adversary $\mathcal{A}$, running in time at most t, making at most q_{Sig} signature queries and at most q_{H} hash oracle queries, that wins the above experiment with probability at least ϵ.

3 Our construction

In this section, we describe our blind signature scheme and prove its security in terms of *blindness* and *one-more unforgeability*. As a warm-up, we recall the GPV signature scheme and trapdoor function from [GPV08]. Then, we show how it can be used to implement blind signatures. For both, trapdoor function and signature scheme, we need a modification that has been clearly marked below. We keep this section "lattice-free" for ease of presentation. The necessary details are given in Section 4.

The underlying family of trapdoor functions is a triple $(\mathsf{TrapGen}, \mathsf{SampleDom}, \mathsf{SamplePre})$, with the following properties.

Function generation. There is an efficient algorithm $\mathsf{TrapGen}$ that outputs $(a, t) \leftarrow \mathsf{TrapGen}(n)$, where a fully defines the function f_a and the trapdoor t is used to sample from the inverse $f_t^{-1}(\cdot)$, which is defined as $\mathsf{SamplePre}(t, \cdot)$.

Efficiency. The function $f_a : D_n \to R_n$ is efficiently computable. Furthermore, the three finite sets R_n, D_n, $D_n^\star$ are efficiently recognizable and R_n is closed under addition. Furthermore, let $D_n^\star \subseteq D_n$, such that $x_1 \pm x_2 \in D_n$ for all $x_1, x_2 \in D_n^\star$. D_n and $D_n^\star$ are *not* closed under addition.

One-wayness. Computing the function $f_t^{-1} : R_n \to D_n^\star$, is infeasible without the trapdoor t.

Domain sampling with uniform output. $\mathsf{SampleDom}(n)$ samples from some distribution over $D_n^\star$, such that their images under f_a are uniformly distributed over R_n.

Preimage sampling. Let $y \in R_n$. $f_t^{-1}(y)$ samples $x \leftarrow \mathsf{SampleDom}(n)$ under the condition that $f_a(x) = y$. The conditional entropy of such values x is at least $\omega(\log(n))$.

Linearity. Let $x_1 + x_2 \in D_n$: $f_a(x_1 + x_2) = f_a(x_1) + f_a(x_2)$.

Collision resistance. There is no algorithm $\mathcal{A}(n, a)$ that outputs a pair $(x, x') \in D_n^2$, such that $x \neq x'$ and $f_a(x) = f_a(x')$, in time polynomial in n with noticeable probability.

Note that we slightly modified the original setting regarding the sets $D_n, D_n^\star$. In [GPV08], it is always the same, whereas we have introduced different $D_n, D_n^\star$ for trapdoor evaluation and preimage sampling, respectively. As in the original work, we will always assume that the above properties, especially the statistical distributions, hold for f_a in a perfect sense.

In addition to the above trapdoor function, Gentry, Peikert, and Vaikuntanathan use the "hash-then-sign" paradigm with a full-domain hash function (cf. [BR93]) $\mathsf{H} \leftarrow \mathcal{H}(n)$, where $\mathsf{H} : \{0,1\}^* \to R_n$ and $\mathcal{H}$ is a family of collision resistant hash functions.

With the modification $D_n^\star \subseteq D_n$, the GPV signature scheme is a tuple $(\mathsf{Kg}, \mathsf{Sig}, \mathsf{Vf})$, where:

Key generation. $\mathsf{GPV.Kg}(n)$ outputs $(a, t) \leftarrow \mathsf{TrapGen}(n)$.

Signature issue. Let $M \in \{0,1\}^*$ be a message. $\mathsf{GPV.Sig}(t, M)$ checks whether M has been signed before and, if so, outputs the same signature. Otherwise, it computes $\sigma \leftarrow f_t^{-1}(\mathsf{H}(M))$, stores (M, σ), and returns $\sigma \in D_n^\star$.

Verification. Given a signature σ. $\mathsf{GPV.Vf}(a, \sigma, M)$ returns 1 iff $\sigma \in D_n$ and $f_a(\sigma) = \mathsf{H}(M)$.

The GPV signature scheme is strongly unforgeable under a chosen message attack. In Section 4, we show that this still holds with our modification.

Using a slight relaxation of the above signature scheme, we construct an equally efficient and provably secure blind signature scheme $\mathsf{BS} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Vf})$ as follows.

Key generation. $\mathsf{BS.Kg}(n)$ outputs $(a, t) \leftarrow \mathsf{TrapGen}(n)$, where a is the public verification key and t is the secret signing key, and sets up a list of already signed messages $L_M = \{0\}$.

Signature protocol. The signature issue protocol for messages $M \in R_n$ is shown in Figure 1. Note that the blind signature scheme is again stateful, i.e. the signer does not sign a blinded message twice and it does not sign $M^\star = 0 \in R_n$ in particular. The result is $\sigma \in D_n^\star$.

Verification. $\mathsf{BS.Vf}(a, \sigma, M)$ outputs 1 iff $\sigma \in D_n$ and $f_a(\sigma) = \mathsf{H}(M)$.

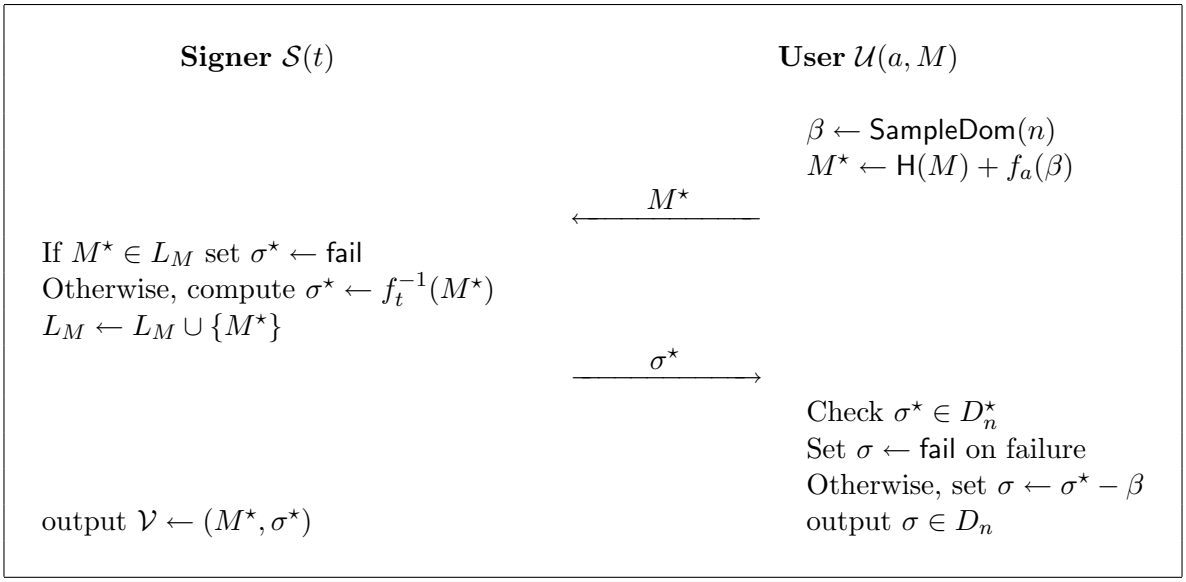

Figure 1: Issue protocol of the blind signature scheme BS

Completeness. The scheme BS is complete because for all honestly generated key pairs (a, t), all messages $m \in \{0,1\}^*$, all outputs $(\beta, M^\star)$ of $\mathsf{BS.Blind}(a, M)$, and all signatures $\sigma^\star \leftarrow \mathsf{BS.Sig}(t, M^\star)$ we have

$$\sigma \leftarrow \sigma^\star - \beta \in D_n$$

and

$$f_a(\sigma) = f_a(\sigma^\star - \beta) = f_a(\sigma^\star) - f_a(\beta) = f_a(f_t^{-1}(\mathsf{H}(M) + f_a(\beta))) - f_a(\beta) = \mathsf{H}(M)\,.$$

Therefore, $\mathsf{BS.Vf}(a, \sigma, M) = 1$.

In the following, we prove the security of our blind signature scheme. A blind signature scheme is secure if it satisfies *blindness* and *one-more unforgeability* as described in Section 2.

Blindness. We prove that, like Chaum's blind signature scheme [Cha83], BS is unconditionally blind, i.e. $(\infty, 0)$-blind. The intuition is that the signer only sees random elements from R_n after the user has applied a random blinding value.

Theorem 3.1 (Blindness) *The blind signature scheme* BS *is* $(\infty, 0)$*-blind.*

The idea of the proof is that, given the signer's views $\mathcal{V}_0, \mathcal{V}_1$ in the experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}$, there are always two equally distributed pairs of blinding values $(f_a(\beta_0), f_a(\beta_1))$ and $(f_a(\beta_0'), f_a(\beta_1'))$, such that both generate the same views, $(f_a(\beta_0), f_a(\beta_1))$ is a witness for b, and $(f_a(\beta_0'), f_a(\beta_1'))$ is a witness for $b' = 1 - b$. From this, we infer that the signer can only guess the correct b with probability $1/2$.

Proof. Let M_0, M_1 be the messages obtained from the signer. Assume that b is fixed and $\beta_0, \beta_1 \leftarrow \mathsf{SampleDom}(n)$ were chosen by the users $\mathcal{U}_0, \mathcal{U}_1$, who compute

$$\begin{aligned} M_0^\star &\leftarrow \mathsf{H}(M_b) + f_a(\beta_0)\,, \\ M_1^\star &\leftarrow \mathsf{H}(M_{1-b}) + f_a(\beta_1)\,, \end{aligned}$$

and receive $\sigma_0^\star$ and $\sigma_1^\star$, such that $\sigma_0^\star$ contains a signature on M_b and $\sigma_1^\star$ contains a signature on M_{1-b}. Then, the signer's views are

$$\begin{aligned} \mathcal{V}_0 &= (M_0^\star, \sigma_0^\star) \\ \text{and } \mathcal{V}_1 &= (M_1^\star, \sigma_1^\star)\,. \end{aligned}$$

Now, we show that for the given choice of blinding values $(f_a(\beta_0), f_a(\beta_1))$, there is exactly one pair (B_0, B_1) that results in the same views while reversing the order, in which the messages are signed. Let

$$\begin{aligned} B_0 &\leftarrow \mathsf{H}(M_b) - \mathsf{H}(M_{1-b}) + f_a(\beta_0) \\ \text{and } B_1 &\leftarrow \mathsf{H}(M_{1-b}) - \mathsf{H}(M_b) + f_a(\beta_1) \end{aligned}$$

and observe that there are $\beta_0', \beta_1' \in D_n^\star$ with $B_0 = f_a(\beta_0')$ and $B_1 = f_a(\beta_1')$. Since H is a random oracle and $f_a(\beta_0)$, $f_a(\beta_1)$ are distributed uniformly at random over R_n, so are the blinding values B_0 and B_1. Assuming $b' = 1 - b$, the blinding values B_0, B_1 yield the following blinded messages:

$$\begin{aligned} \mathcal{U}_0 \text{ computes } M_0^{\star\prime} &\leftarrow \mathsf{H}(M_{b'}) + B_0 = \mathsf{H}(M_{1-b}) + B_0 = \mathsf{H}(M_b) + f_a(\beta_0) = M_0^\star\,; \\ \mathcal{U}_1 \text{ computes } M_1^{\star\prime} &\leftarrow \mathsf{H}(M_{1-b'}) + B_1 = \mathsf{H}(M_b) + B_1 = \mathsf{H}(M_{1-b}) + f_a(\beta_1) = M_1^\star\,. \end{aligned}$$

The resulting views $\mathcal{V}_0'$ and $\mathcal{V}_1'$ of the signer are equal to $\mathcal{V}_0$ and $\mathcal{V}_1$, respectively. Therefore, there are indistinguishable witnesses $(f_a(\beta_0), f_a(\beta_1))$ for b and $(f_a(\beta_0'), f_a(\beta_1'))$ for $b' = 1-b$, which concludes the proof. □

One-more unforgeability. We prove that our blind signature scheme is unforgeable under a reasonable assumption, namely that the following "one-more trapdoor inversion problem" is hard.

Definition 3.2 (Chosen target trapdoor inversion problem (CTTI)) *The chosen target trapdoor inversion problem is defined via the following experiment, where the adversary $\mathcal{A}$ has access to a challenge oracle O_{R_n} and to an inversion oracle f_t^{-1}. The adversary wins, if it outputs $\jmath$ preimages for challenges obtained from O_{R_n}, while making only $\imath < \jmath$ queries to f_t^{-1}. The oracle f_t^{-1} does not answer queries twice and its does not invert $0 \in R_n$.*

Experiment $\mathsf{Exp}_{\mathcal{A}}^{ctti}(n)$
  $(a, t) \leftarrow \mathsf{TrapGen}(n)$
  $(\pi, x_1, \ldots, x_\jmath) \leftarrow \mathcal{A}^{\mathsf{O}_{R_n}, f_t^{-1}(\cdot)}(n, a)$
  Let $y_1, \ldots, y_\ell$ *be the challenges returned by* O_{R_n}.
  Let $\imath$ *be the number of queries to* f_t^{-1}.
    Return 1 *iff*
      1. $\pi : \{1, \ldots, \jmath\} \to \{1, \ldots, \ell\}$ *is injective and*
      2. $x_i \in D_n$ *and* $f_a(x_i) = y_{\pi(i)}$ *for all* $i = 1, \ldots, \jmath$ *and*
      3. $\imath < \jmath$.

The problem is $(t, q_{\mathsf{I}}, q_{\mathsf{O}}, \epsilon)$-hard if there is no algorithm $\mathcal{A}$, running in time at most t, making at most q_{I} inversion queries, and at most q_{O} queries to O_{R_n}, which wins the above experiment with probability at least ϵ. The one-wayness of f_a gives us $(\mathsf{poly}(n), 0, 1, \epsilon)$-hardness, which we will extend to $(\mathsf{poly}(n), \mathsf{poly}(n), \mathsf{poly}(n), \epsilon')$-hardness for a negligible ϵ'. With our definition and this

assumption, we follow the line of thought of Bellare, Namprempre, Pointcheval, and Semanko in [BNPS03]. They define a collection of "one-more" problems in the RSA context, which are perfectly tailored for proving one-more unforgeability. In [BMV08], Bresson, Monnerat, and Vergnaud give a separation result on these "one-more" problems, showing that they cannot be proven equivalent to "simple" RSA inversion. The same seems to apply here as we are not able to prove equivalence of CTTI and simple trapdoor inversion.

In the following, we will assume $(\mathsf{poly}(n), \mathsf{poly}(n), \mathsf{poly}(n), \epsilon)$-hardness of CTTI on the grounds that it is directly related to the provably hard problem of forging GPV signatures. In both cases, one has to find a solution $x \in D_n$ to the equation $f_a(x) = y$ for a given y, while knowing polynomially many distinct preimage-image pairs. Furthermore, we argue that an adversary cannot successfully recombine answers of f_t^{-1} in order to produce more preimages than there were oracle queries because adding or subtracting more than two of the oracle's answers yields an invalid preimage $\notin D_n$ with high probability because D_n is not closed under addition. This is stated more formally in Section 4.

Theorem 3.3 (One-more unforgeability) *The* BS *blind signature scheme is* $(t, q_{\mathsf{Sig}}, q_{\mathsf{H}}, \epsilon)$*-one-more unforgeable if the* CTTI *is* $(t, q_{\mathsf{Sig}}, q_{\mathsf{H}}, \epsilon)$*-hard.*

Proof. Towards contradiction, we assume that there exists a successful forger $\mathcal{A}$ against one-more unforgeability of BS. Using $\mathcal{A}$, we construct an algorithm $\mathcal{B}$ via a black-box simulation, such that $\mathcal{B}$ solves the respective instance of the CTTI. The simulation works as follows.

Setup. $\mathcal{B}$ gets as input the public trapdoor parameter a and has access to the challenge oracle O_{R_n} and to a trapdoor inversion oracle f_t^{-1}. $\mathcal{B}$ initializes a list $L_{\mathsf{H}} \leftarrow \emptyset$ of pairs (M, c), indexed by M, a list $L_{\mathsf{I}} \leftarrow \emptyset$ of pairs $(M^\star, \sigma^\star)$, indexed by $M^\star$, and two counters $\ell \leftarrow 0$, $\imath \leftarrow 0$. It runs $\mathcal{A}$ on input a in a black-box simulation.

Random oracle queries. On input M, $\mathcal{B}$ looks up M in L_{H}. If it finds a pair (M, c) then it returns c. Otherwise, $\mathcal{B}$ increments $\imath$, chooses a new $c_\imath \leftarrow \mathsf{O}_{R_n}$, stores $(M_\imath \leftarrow M, c_\imath)$ in L_{H}. Afterwards, $\mathcal{B}$ returns $c_\imath$.

Blind signature queries. On input $M^\star$, algorithm $\mathcal{B}$ searches a pair $(M^\star, \sigma^\star)$ in L_{I}. If it exists, $\mathcal{B}$ returns $\sigma^\star$. Otherwise, algorithm $\mathcal{B}$ increments ℓ, queries its inversion oracle $\sigma_\ell^\star \leftarrow f_t^{-1}(M^\star)$, stores $(M_\ell^\star \leftarrow M^\star, \sigma_\ell^\star)$ in L_{I}, and returns $\sigma_\ell^\star$.

Output. Finally, $\mathcal{A}$ stops and outputs $((M_1, \sigma_1), \ldots, (M_\jmath, \sigma_\jmath))$, $\ell < \jmath$, for distinct messages. W.l.o.g., assume that $(M_i, c_i) \in L_{\mathsf{H}}$, for all $i = 1, \ldots, \jmath$. Algorithm $\mathcal{B}$ sets

$$\pi = \{(i, j) : f_a(\sigma_i) = c_j\}$$

and outputs $(\pi, \sigma_1, \ldots, \sigma_\jmath)$.

Analysis. First, observe that all of $\mathcal{A}$'s oracles are perfectly simulated. When $\mathcal{A}$ calls H, algorithm $\mathcal{B}$ draws a new challenge from its challenge oracle. Whenever $\mathcal{A}$ queries its signature oracle on a new blinded message, $\mathcal{B}$ calls its inversion oracle. Therefore, when $\mathcal{A}$ outputs a one-more forgery, $\mathcal{B}$ can use it to solve the CTTI. $\mathcal{B}$'s output is valid in the CTTI experiment because all preimages $(\sigma_i \in D_n)$ evaluate to challenges received from O_{R_n} and the number of output inversions $\jmath$ is greater than the number of inversion queries ℓ. As for the map π, we state that it is injective. Otherwise, there would be a pair $\sigma \neq \sigma'$ in $\mathcal{A}$'s output with $f_a(\sigma) = f_a(\sigma') = \mathsf{H}(M_i)$, which contradicts the collision resistance of f_a. Thus, $\mathcal{B}$ is successful if $\mathcal{A}$ is. □

4 Security of the modified GPV signature scheme

In order to provide a modified security proof for the modified GPV signature scheme with an enlarged signature domain, we need the following facts from lattice theory. Afterwards, we adapt the proof from [GPV08] to our setting.

4.1 Lattices

A lattice in $\mathbb{R}^n$ is a set $\Lambda = \{\sum_{i=1}^{d} x_i \, \mathbf{b}_i \,|\, x_i \in \mathbb{Z}\}$, where $\mathbf{b}_1, \ldots, \mathbf{b}_d$ are linearly independent over $\mathbb{R}$. The matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_d]$ is a *basis* of the lattice Λ and we write $\Lambda = \Lambda(\mathbf{B})$. The number of linearly independent vectors in the basis is the dimension of the lattice. Now, consider *modular lattices* as a special form of lattices. Given a modulus q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and the equation

$$\mathbf{A}\,\mathbf{v} \equiv \mathbf{0} \pmod{q},$$

then the set of all vectors $\mathbf{v} \in \mathbb{Z}_q^m$ that satisfy the above equation is a lattice. Lattices of this form are denoted with $\Lambda_q^{\perp}(\mathbf{A})$.

The main computational problem in lattices is the (approximate) shortest vector problem (SVP), where an algorithm is given a description, a basis, of a lattice Λ and is supposed to find the shortest vector $\mathbf{v} \in \Lambda \backslash \{\mathbf{0}\}$ with respect to a certain ℓ_p norm (up to an approximation factor). More precisely, find a vector $\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}$, such that

$$\|\mathbf{v}\|_p \leq \gamma \, \|\mathbf{w}\|_p \text{ for all } \mathbf{w} \in \Lambda \setminus \{\mathbf{0}\}$$

for a fixed approximation factor $\gamma \geq 1$. This problem is known to be $\mathcal{NP}$-hard for all ℓ_p norms [Din02, RR06, Kho05] with a constant approximation factor. For exponential (in the lattice dimension) approximation factors, the problem is solvable in polynomial time by the famous LLL algorithm by Lenstra, Lenstra, and Lovász [LLL82]. We refer the interested reader to a recent survey [Reg07] by Regev for the currently known "approximability" and "inapproximability" results. The practical hardness of these lattice problems is analyzed in [BLR08].

In the special case of modular lattices, there is also a special version of the SVP, named short integer solution problem (SIS). There, an algorithm is given a basis of $\Lambda_q^{\perp}(\mathbf{A})$ and is supposed to output a non-zero solution $\mathbf{v} \in \mathbb{Z}_q^m$ to the above equation. The algorithm succeeds if $\|\mathbf{v}\|_p \leq \nu$ for a given norm bound ν. The SIS was, in principle, introduced by Ajtai [Ajt96] and its hardness is analyzed in [MR07] and [GPV08]. The latter work also explicitly deals with the ℓ_∞ norm, which we will use in our security proofs. We write $\mathsf{SIS}^p(m, q, \nu)$ for the SIS problem in m-dimensional lattices $\Lambda_q^{\perp}(\mathbf{A})$ with norm bound ν w.r.t. the ℓ_p norm. The problem is (t, ϵ)-hard if no algorithm that runs in time t can solve it with probability at least ϵ.

4.2 Proof of security

Using lattices, the GPV trapdoor is implemented as a matrix-vector multiplication $\mathbf{A}\mathbf{x}$ modulo q, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $m = \mathcal{O}(n \log(n))$, and $q = \mathcal{O}(n^3)$. Furthermore, we have $R_n = \mathbb{Z}_q^n$,

$$D_n = \{\mathbf{x} \in \mathbb{Z}^m : \|\mathbf{x}\|_\infty \leq 2D\},$$

and

$$D_n^\star = \{\mathbf{x} \in \mathbb{Z}^m : \|\mathbf{x}\|_\infty \leq D\},$$

where D is about $n\log^2(n)$. We assume that f_t^{-1} returns preimages in $D_n^\star$, whereas signatures may be in D_n. Concerning security of the GPV signature scheme, [GPV08] states that it is strongly unforgeable if $\mathsf{SIS}^\infty(m,q,2D)$ is hard. As for our modified setting, with $D_n^\star \neq D_n$, we need a slightly stronger assumption, i.e. $\mathsf{SIS}^\infty(m,q,3D)$ has to be hard. Furthermore, we claim that this special setting cannot be exploited to forge a signature $\sigma' \in D_n$ from two valid signatures $\sigma_1, \sigma_2 \in D_n^\star$ by simply adding them as $\sigma' \leftarrow \sigma_1 + \sigma_2$ because of the collision resistance of the full-domain hash H. More formally, think of a closure C under addition (mod q) on D_n, given a set of signaturues $\sigma_1, \ldots, \sigma_Q \in D_n$. Furthermore, assume that there are approximately m signatures per hash value $\mathsf{H}(M) \in R_n$ (cf. [GPV08]). Then, the probability of hitting one of those signatures in D_n with a value from C is at most $m|C|/|D_n|$. Observe that $|C|$ has to stay polynomial in n for the adversary to be efficient and $m = \mathcal{O}(n\log(n))$. The magnitude of D_n is, however, exponential in n, which makes such adversaries fail but with negligible probability.

We support this reasoning by adapting the the proof from [GPV08] (Proposition 6.1 in the extended version).

Theorem 4.1 *Let the parameters $n, m, q, \Lambda_q^\perp(\mathbf{A})$ be as defined above and let $T_{\mathsf{SampleDom}}(n), T_{f_a}(n)$ be the cost functions for domain sampling and trapdoor evaluation. The modified GPV signature scheme is $(t, q_{\mathsf{Sig}}, q_{\mathsf{H}}, \epsilon)$-strongly unforgeable if $\mathsf{SIS}^\infty(m,q,3D)$ is (t',ϵ')-hard, where $t' = t + (q_{\mathsf{Sig}} + q_{\mathsf{H}})\,(T_{\mathsf{SampleDom}}(n) + T_{f_a}(n))$ and $\epsilon' = \epsilon - 2^{-\omega(\log(n))}$.*

Proof. Given a successful adversary $\mathcal{A}$ against strong unforgeability with success probability ϵ, we build an algorithm $\mathcal{B}$ that finds a collision in f_a and, with that, a short vector in $\Lambda_q^\perp(\mathbf{A})$. Algorithm $\mathcal{B}$ runs $\mathcal{A}$ in a black-box simulation and uses random oracle techniques to simulate the signature oracle.

Setup. Algorithm $\mathcal{B}$ receives the public parameter a of $\Lambda_q^\perp(\mathbf{A})$ as input sets up a list $L_{\mathsf{H}} \leftarrow \emptyset$ of triples (M, h, σ) in order to simulate H and f_t^{-1} consistently. It runs $\mathcal{A}$ on input a.

Random oracle H. When queried with $M \in \{0,1\}^*$, algorithm $\mathcal{B}$ looks for a triple $(M, h, \sigma) \in L_{\mathsf{H}}$. If it exists, $\mathcal{B}$ returns h. Otherwise, the simulator chooses $\sigma \leftarrow \mathsf{SampleDom}(n)$, sets $h \leftarrow f_a(\sigma)$, stores (M, h, σ) in L_{H}, and returns h.

Signature queries. On input $M \in \{0,1\}^*$, algorithm $\mathcal{B}$ runs $\mathsf{H}(M)$, yielding a triple $(M, h, \sigma) \in L_{\mathsf{H}}$. The simulator returns $\sigma \in D_n^\star$.

Output. Finally, $\mathcal{A}$ stops and returns a valid forgery (M^*, σ^*) with $\mathsf{H}(M^*) = h^*$ and $\sigma^* \in D_n$. W.l.o.g., there is a triple $(M^*, h^*, \sigma) \in L_{\mathsf{H}}$ with $\sigma \in D_n^\star$. Algorithm $\mathcal{B}$ outputs $\sigma^* - \sigma$.

Analysis. Observe that $\mathcal{B}$ simulates the random oracle and the signature oracle perfectly and consistently. As for the output of $\mathcal{B}$, we have to show that $\sigma^* - \sigma \neq \mathbf{0}$ holds but with negligible probability. We have to distinguish three cases:

1. If $\sigma^* \in D_n \setminus D_n^\star$, the condition trivially holds.
2. The adversary $\mathcal{A}$ outputs a forgery in the strong sense, i.e. it has previously queried its signature oracle on M^*. Then, we have $\sigma^* - \sigma \neq \mathbf{0}$ by definition.

3. Algorithm $\mathcal{A}$ has not queried its signature oracle on M^*. W.l.o.g., it has queried H on M^* and $\mathcal{B}$ has a triple $(M^*, h^*, \sigma) \in L_{\mathsf{H}}$. By the minimum conditional entropy $\omega(\log(n))$ of σ, we infer that $\sigma^* = \sigma$ with probability at most $2^{-\omega(\log(n))}$, which is still negligible.

Since σ^* and σ are valid signatures on M^*, we have

$$f_a(\sigma^*) = \mathsf{H}(M^*) = f_a(\sigma)$$

and therefore a non-trivial solution to the characteristic equation of $\Lambda_q^{\perp}(\mathbf{A})$:

$$\mathbf{A}(\sigma^* - \sigma) \equiv \mathbf{0} \bmod q\,.$$

In consequence, algorithm $\mathcal{B}$ has learned a lattice vector of norm

$$\|\sigma^* - \sigma\|_\infty \leq 2D + D \leq 3D\,.$$

The overhead of the reduction is dominated by the computational cost for domain sampling and trapdoor evaluation. In the worst-case, the adversary $\mathcal{A}$ never queries H and Sig with the same message, which is why the overhead is $(q_{\mathsf{Sig}} + q_{\mathsf{H}})\,(T_{\mathsf{SampleDom}}(n) + T_{f_a}(n))$. □

5 Conclusions

We have shown how to construct a provably secure blind signature scheme from a lattice based signature scheme in the random oracle model, which can replace current schemes to achieve post-quantum security. Furthermore, as our basis of security, we have introduced the chosen target trapdoor inversion problem (CTTI) that may be useful for other applications as well.

References

[Abe01] Masayuki Abe. *A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures.* Advances in Cryptology — Eurocrypt 2001, Lecture Notes in Computer Science, pages 136–151. Springer-Verlag, 2001.

[Ajt96] Miklós Ajtai. *Generating Hard Instances of Lattice Problems (Extended Abstract).* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1996, Lecture Notes in Computer Science, pages 99–108. Springer-Verlag, 1996.

[BLR08] Johannes Buchmann, Richard Lindner, and Markus Rückert. *Explicit hard instances of the shortest vector problem.* Post-Quantum Cryptography — PQCrypto 2008, Lecture Notes in Computer Science, pages 79–94. Springer-Verlag, 2008.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. *Separation Results on the "One-More" Computational Problems.* Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA)2008, Lecture Notes in Computer Science, pages 71–87. Springer-Verlag, 2008.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. *The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. Journal of Cryptology*, 16(3):185–215, 2003.

[Bol03] Alexandra Boldyreva. *Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme.* Public-Key Cryptography (PKC) 2003, Volume 2567 of Lecture Notes in Computer Science, pages 31–46. Springer-Verlag, 2003.

[BR93] Mihir Bellare and Pil Rogaway. *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.* Proceedings of the Annual Conference on Computer and Communications Security (CCS). ACM Press, 1993.

[Cha83] David Chaum. *Blind Signatures for Untraceable Payments.* Advances in Cryptology — Crypto 1982, pages 199–203. Plemum, New York, 1983.

[CKW04] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. *Efficient Blind Signatures Without Random Oracles.* Security in Communication Networks, Volume 3352 of Lecture Notes in Computer Science, pages 134–148. Springer-Verlag, 2004.

[Din02] Irit Dinur. *Approximating SVP_∞ to within almost-polynomial factors is NP-hard. Theoretical Computer Science*, 285(1):55–71, 2002.

[Fis06] Marc Fischlin. *Round-Optimal Composable Blind Signatures in the Common Reference String Model.* Advances in Cryptology — Crypto 2006, Volume 4117 of Lecture Notes in Computer Science, pages 60–77. Springer-Verlag, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. *Trapdoors for hard lattices and new cryptographic constructions.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2008, pages 197–206. ACM Press, 2008.

[HKKL07] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. *Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions.* Theory of Cryptography Conference (TCC) 2007, Volume 4392 of Lecture Notes in Computer Science, pages 323–341. Springer-Verlag, 2007.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. *Security of Blind Digital Signatures.* Advances in Cryptology — Crypto 1997, Volume 1294 of Lecture Notes in Computer Science, pages 150–164. Springer-Verlag, 1997.

[Kho05] Subhash Khot. *Hardness of approximating the shortest vector problem in lattices. J. ACM*, 52(5):789–808, 2005.

[KZ05] Aggelos Kiayias and Hong-Sheng Zhou. *Two-Round Concurrent Blind Signatures without Random Oracles.* Number 2005/435 in Cryptology eprint archive. `eprint.iacr.org`, 2005.

[LLL82] A. Lenstra, H. Lenstra, and L. Lovász. *Factoring polynomials with rational coefficients. Mathematische Annalen*, 261(4):515–534, 1982.

[LMPR08] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. *SWIFFT: A Modest Proposal for FFT Hashing.* Fast Software Encryption (FSE) 2008, Lecture Notes in Computer Science, pages 54–72. Springer-Verlag, 2008.

[MR07] Daniele Micciancio and Oded Regev. *Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing*, 37(1):267–302, 2007.

[Oka06] Tatsuaki Okamoto. *Efficient Blind and Partially Blind Signatures Without Random Oracles.* Theory of Cryptography Conference (TCC) 2006, Volume 3876 of Lecture Notes in Computer Science, pages 80–99. Springer-Verlag, 2006.

[PS97] David Pointcheval and Jacques Stern. *New Blind Signatures Equivalent to Factorization (extended abstract).* ACM Conference on Computer and Communications Security, pages 92–99. ACM Press, 1997.

[PS00] David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology*, 13(3):361–396, 2000.

[Reg07] Oded Regev. *On the Complexity of Lattice Problems with Polynomial Approximation Factors*, 2007. A survey for the LLL+25 conference.

[RR06] Oded Regev and Ricky Rosen. *Lattice problems and norm embeddings.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2006, pages 447–456. ACM Press, 2006.

[Sho97] Peter W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing*, 26(5):1484–1509, 1997.