Efficient Quantum-immune Blind Signatures
— preliminary version —

Markus Rückert
rueckert@cdc.informatik.tu-darmstadt.de

Cryptography and Computeralgebra
Department of Computer Science
TU Darmstadt

September 15, 2008

Keywords Quantum-immune, blind signatures, lattice

Abstract We present the first quantum-immune blind signature scheme. Our scheme is provably secure, efficient, and round-optimal. Its basis of security is the problem of finding short vectors in a modular lattice.

1 Introduction

Since 1982, when David Chaum proposed his idea of blind signatures and a, by now classic, application in the context of digital payments, numerous blind signature schemes and other privacy-enhanced signature schemes have been developed. According to the security model, mainly influenced by Juels, Luby, and Ostrovsky [14] as well as Pointcheval and Stern [20], 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 signature than there were interactions with the signer.

Today, when building provably secure signature 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.

To date, there are only a few cryptographic assumptions that are conjectured to be *quantum-immune*, i.e. they are considered to be able to withstand quantum computer attacks. One of those assumptions is the hardness of approximating shortest vectors (SVP) in a lattice. Although the works of Ludwig [16] and Regev [21] suggest that todays lattice reduction algorithms can benefit from the intrinsic parallelity in quantum computation, this does not invalidate the assumption. Slightly larger security parameters appear to be a sufficient countermeasure.

Our Contribution and related work. Using the SVP as our security assumption, we construct the first quantum-immune blind signature scheme. As for its efficiency, we state that it is almost as efficient as the underlying signature scheme proposed by Gentry, Peikert, and Vaikuntanathan (GPV) [13] and with its two rounds, it is even *round-optimal.* The security of both, GPV signature scheme and our blind signature scheme, is proven in the random oracle model and, due to Ajtai's result, is based on the worst case hardness of the SVP.

All previous (efficient) constructions [11,19,20,1,7,12,15,18] have one thing in common. They are built upon number theoretic assumptions, like the hardness of factoring large integers or computing discrete logarithms. The newer approaches of Boldyreva [8] and Okamoto [18] 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 powerful quantum computers, where both factoring and computing discrete logarithms becomes easy due to the seminal work of Peter Shor [22].

Despite the uninstantiability result of Canetti, Goldreich, and Halevi [10], we believe that our construction is an important step towards quantum-immune blind signature schemes.

Organization. After a brief 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 details and the realization of the underlying trapdoor permutation.

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 $\mathcal{B}$.

Digital signatures. Let's recall the definitions of digital signature schemes and of blind signatures schemes. 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 Generation. $\mathsf{Sig}(sk, m)$ outputs a signature σ on a message m from the message space $\mathcal{M}$ under sk.

Signature Verification. The algorithm $\mathsf{Vf}(pk, \sigma, m)$ outputs 1 iff σ is a valid signature on m under pk.

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

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 our construction, 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. The described concept is formalized in following experiment.

Experiment $\mathsf{Exp}^{\mathsf{su\text{-}cma}}_{\mathcal{A},\mathsf{DS}}(n)$
$(sk, pk) \leftarrow \mathsf{Kg}(n)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sig}(sk,\cdot)}(pk)$
let (m_i, σ_i) be the answer returned by $\mathsf{Sig}(sk, \cdot)$ on input m_i, for $i = 1, \ldots, k$.
Return 1 iff $\mathsf{Vf}(pk, m^*, \sigma^*) = 1$ and $(m^*, \sigma^*) \notin \{(m_1, \sigma_1), \ldots, (m_k, \sigma_k)\}$.

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

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 Generation. $\mathsf{Sig}(sk, m)$ describes the joint execution of $\mathcal{S}$ and $\mathcal{U}$. The private output of the $\mathcal{S}$ is a view $\mathcal{V}$ and the private output of the $\mathcal{U}$ is a signature on the message m under sk. Thus, we write $(\mathcal{V}, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m)\rangle$.
Signature Verification. $\mathsf{Vf}(pk, \sigma, m)$ outputs 1 iff σ is a valid signature on m under pk.

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 noticable probability.

As for security, blind signatures have to satisfy two properties: blindness and one-more unforgeability [14,20]. The notion of blindness is defined in the following experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}$, 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. After seeing the unblinded signatures in the original order, according to m_0, m_1, the signer has to guess the message that has been signed for the first user.

Experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}(n)$
$b \leftarrow \{0, 1\}$
$(pk, sk) \leftarrow \mathsf{BS.Kg}(n)$
$(m_0, m_1) \leftarrow \mathcal{S}^*(pk, sk)$
$(\mathcal{V}_0, \sigma_b) \leftarrow \langle \mathcal{S}^*(sk), \mathcal{U}_0(pk, m_b)\rangle$
$(\mathcal{V}_1, \sigma_{1-b}) \leftarrow \langle \mathcal{S}^*(sk), \mathcal{U}_1(pk, m_{1-b})\rangle$
Either of the signatures might equal fail.
If $\sigma_0 \neq \mathsf{fail}$ and $\sigma_1 \neq \mathsf{fail}$
  $d \leftarrow \mathcal{S}^*(sk, pk, \sigma_0, \sigma_1)$
Else
  $d \leftarrow \mathcal{S}^*(sk, pk, \mathsf{fail}, \mathsf{fail})$
Return 1 iff $d = b$

Now consider a second experiment $\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}$, where both users are controlled by a simulator $\mathcal{SIM}$. The order, in which the signatures are obtained, is reversed here. Thus, if the simulator's behaviour is indistinguishable from that of two regular users, a successful adversary $\mathcal{S}^*$ in $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}$ outputs the guess $d = b$ in $\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}$ and succeeds.

Experiment $\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}(n)$
$b \leftarrow \{0, 1\}$
$(pk, sk) \leftarrow \mathsf{BS.Kg}(n)$
$(m_0, m_1) \leftarrow \mathcal{S}^*(pk, sk)$
Setup $\mathcal{SIM}(n, pk, b, m_0, m_1)$
$(\mathcal{V}_0, \sigma_{1-b}) \leftarrow \langle \mathcal{S}^*(sk), \mathcal{SIM}(\mathsf{run}_0)\rangle$
$(\mathcal{V}_1, \sigma_b) \leftarrow \langle \mathcal{S}^*(sk), \mathcal{SIM}(\mathsf{run}_1)\rangle$
Either of the signatures might equal fail.
If $\sigma_0 \neq \mathsf{fail}$ and $\sigma_1 \neq \mathsf{fail}$

$d \leftarrow \mathcal{S}^*(sk, pk, \sigma_0, \sigma_1)$
Else
$d \leftarrow \mathcal{S}^*(sk, pk, \mathsf{fail}, \mathsf{fail})$
Return 1 iff $d = b$

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

$$\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}} = \left| \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}(n) = 1] - \Pr[\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}(n) = 1] \right| .$$

The second security property, one-more unforgeability, ensures that each 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 $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(n)$
$(pk, sk) \leftarrow \mathsf{BS.Kg}(n)$
$\{(m_1, \sigma_1), \ldots, (m_\jmath, \sigma_\jmath)\} \leftarrow \mathcal{U}^{*\mathsf{H}(\cdot),\langle \mathcal{S}(sk),\cdot \rangle}(pk)$
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 $1 \leq i < j \leq \jmath$
2. $\mathsf{BS.Vf}(pk, \sigma_i, m_i) = 1$ 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 the construction of our blind signature scheme and prove its security in terms of *blindness* and *one-more unforgeability*.

The underlying signature scheme was developed by Gentry, Peikert, and Vaikuntanathan (GPV) and presented at STOC 2008 [13]. It is built upon a family of trapdoor functions, which are almost as good as trapdoor permutations. The family is described via a triple ($\mathsf{TrapGen}$, $\mathsf{SampleDom}$, $\mathsf{SamplePre}$) and has, among others, 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 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 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 $x_1, x_2 \in D_n^\star$

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 values from some distribution over $D_n^\star$, such that their images under f_a are uniformly distributed over R_n.

Pre-image 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 entropy of x is to be 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 exists 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 noticable probability.

Note that we slightly modified the original construction regarding the domain D_n. In [13], it is always the same, whereas we have introduced a different $D_n^\star$ for preimage sampling. 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. [9]) $\mathsf{H} \leftarrow \mathcal{H}(n)$, where $\mathsf{H} : \{0,1\}^* \to R_n$ and $\mathcal{H}$ is a family of collision-resistant hash functions. In this setting, the GPV signature scheme is strongly unforgeable under a chosen message attack (cf. [13]). In Section 4, we show that this still holds with our modification.

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

Key generation. $\mathsf{GPV.Kg}(1^n)$ outputs $(a, t) \leftarrow \mathsf{TrapGen}(1^n)$.

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 $f_a(\sigma) = \mathsf{H}(m)$.

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

Signature protocol. The signature issue protocol for a message $m \in \{0,1\}^*$ is shown in Figure 1.

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

If the user outputs fail, it may be that the signer is dishonest. In the special setting of e-cash, if the obtained signature σ is not in the domain of f_a, the process has to be repeated with a different m and the receiver of the signature has to reveal β to prove to the signer that the signature is literally worthless. For the moment, we assume that $\sigma^\star - \beta \in D_n$. In Section 4, its becomes obvious that this always holds if both parties are honest.

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 called secure if it satisfies *blindness* and *one-more unforgeability* as described in Section 2.

Signer $\mathcal{S}(t)$		User $\mathcal{U}(a, m)$
		$\beta \leftarrow \mathsf{SampleDom}(n)$ $m^\star \leftarrow \mathsf{H}(m) + f_a(\beta)$
	$\xleftarrow{\quad m^\star \quad}$	
$\sigma^\star \leftarrow f_t^{-1}(m^\star)$		
	$\xrightarrow{\quad \sigma^\star \quad}$	
		Check $\sigma^\star \in D_n^\star$ Set $\sigma \leftarrow \mathsf{fail}$ on failure Otherwise, set $\sigma \leftarrow \sigma^\star - \beta$
output $\mathcal{V} \leftarrow (m^\star, \sigma^\star)$		output $\sigma \in D_n$

Fig. 1. Issue protocol of the blind signature scheme BS

Blindness. We prove that, like Chaum's blind signature scheme [11], BS is unconditionally blind, i.e. $(\infty, 0)$-blind. In short, we construct a simulator that, given any b, any two messages, and any pair (β_0, β_1), constructs a pair of blinding values that generate the same view (as with b), while reversing the order in which the signatures are obtained from the signer (as with $1 - b$).

Theorem 1 (Blindness). *The blind signature scheme* BS *is* $(\infty, 0)$*-blind.*

Proof. We define a simulator $\mathcal{SIM}$ that, given a fixed b, executes the signature protocol in run_0 as

$$(\mathcal{V}_0, \sigma_{1-b}) \leftarrow \langle \mathcal{S}^*(t), \mathcal{SIM}(\mathsf{run}_0) \rangle$$

and in run_1 as

$$(\mathcal{V}_1, \sigma_b) \leftarrow \langle \mathcal{S}^*(t), \mathcal{SIM}(\mathsf{run}_1) \rangle \ .$$

Form $\mathcal{S}^*$'s point of view, the simulated users are indistinguishable from honest users. Thus, the output of $\mathcal{S}^*$ is the same in both experiments while the order, in which the messages are signed, is effectively reversed. The simulator works as follows:

Setup. $\mathcal{SIM}$ gets as input a bit b, two messages m_0, m_1, and the public parameters of BS. $\mathcal{SIM}$ chooses

$$\beta_0, \beta_1 \leftarrow \mathsf{SampleDom}(n)$$

and sets

$$B'_0 \leftarrow \mathsf{H}(m_{1-b}) - \mathsf{H}(m_b) + f_a(\beta_1)$$
$$B'_1 \leftarrow \mathsf{H}(m_b) - \mathsf{H}(m_{1-b}) + f_a(\beta_0) \, .$$

Mode run_0. Set $m^\star \leftarrow \mathsf{H}(m_b) + B'_0$. Execute the protocol $(\mathcal{V}_0, \sigma_{1-b}) \leftarrow \langle \mathcal{S}^*(t), \mathcal{SIM}(\mathsf{run}_0) \rangle$, where σ_{1-b} is computed as $\sigma^\star - \beta_1$.

Mode run_1. Set $m^\star \leftarrow \mathsf{H}(m_{1-b}) + B'_1$. Execute the protocol $(\mathcal{V}_1, \sigma_b) \leftarrow \langle \mathcal{S}^*(t), \mathcal{SIM}(\mathsf{run}_1) \rangle$, where σ_b is computed as $\sigma^\star - \beta_0$.

Output. The two signatures σ_0, σ_1.

Analyis. In short, for each choice of b and blinding values $f_a(\beta_0), f_a(\beta_1)$, there is exactly one pair of blinding values B'_1, B'_0 that generates a views for $b' = 1 - b$ that are equal to the views with b.

The choice of B'_0, B'_0 is fully random due to the fact that $f_a(\beta_0)$ and $f_a(\beta_1)$ are distributed uniformly at random over R_n and because H is a random oracle. Observe that for B'_0 and B'_1, there exist $\beta'_0, \beta'_1 \in D^\star_n$, such that $f_a(\beta'_0) = B'_0$, $f_a(\beta'_1) = B'_1$. Therefore, these blinding values look exactly like the blinding values chosen by two honest users.. Because of that, the signer's views are indistinguishable in experiments $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}$ and $\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}$. In consequence, the signer's output d is correct in both experiments. The simulator, however, obtains the signatures in reverse order w.r.t. the honest users, i.e. first for m_{1-b} and then for m_b. Hence,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}(n) &= \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}(n) = 1] - \Pr[\mathsf{Exp}^{\mathsf{blind\text{-}sim}}_{\mathcal{S}^*,\mathcal{SIM},\mathsf{BS}}(n) = 1] \\
&= \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}(n) = 1] - \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathcal{U}_0,\mathcal{U}_1,\mathsf{BS}}(n) = 1] = 0\,.
\end{aligned}$$

□

One-more unforgeability. We prove that our blind signature scheme is provably secure under a reasonable assumption, namely that the following "one-more trapdoor inversion problem" is hard. With our definition and this assumption, we follow the line of thought of Bellare, Namprempre, Pointcheval, and Semanko in [7]. They define a collection of "one-more" problems in the RSA context.

Definition 1 (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}.*

Experiment $\mathsf{Exp}^{ctti}_{\mathcal{A}}(n)$
$(a, t) \leftarrow \mathsf{TrapGen}(n)$
$(\pi, x_1, \ldots, x_\jmath) \leftarrow \mathcal{A}^{\mathsf{O}_{R_n}, f_t^{-1}(\cdot)}(n, a)$
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. $f_a(x_i) = y_{\pi(i)}$ *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'.

As for its hardness, we show that breaking CTTI is as hard as forging GPV signatures.

Theorem 2. *The* CTTI *is* $(t, q_\mathsf{I}, q_\mathsf{O}, \epsilon)$*-hard if and only if the* GPV *signature is* $(t, q_\mathsf{I}, q_\mathsf{O}, \epsilon)$*-strongly unforgeable.*

Proof. We show both directions separately.

$\mathsf{CTTI} \Rightarrow \mathsf{GPV}$*:* Let's assume that GPV is not $(t, q_{\mathsf{I}}, q_{\mathsf{O}}, \epsilon)$-strongly unforgeable. Thus, there exists a forger $\mathcal{A}$ against strong unforgeability. Using $\mathcal{A}$, we construct an adversary $\mathcal{B}$ that solves the CTTI. The adversary $\mathcal{B}$ works as follows.

Setup. $\mathcal{B}$ sets up a list $L_{\mathsf{H}} \leftarrow \emptyset$ of triples (m, c, s), which is indexed by the first component, and a counter $\ell \leftarrow 0$. It gets as input the public trapdoor parameter a and executes $\mathcal{A}$ on input a in a black-box simulation. $\mathcal{B}$ has access to O_{R_n} and $f_t^{-1}(\cdot)$.

Random oracle H. For each query m of $\mathcal{A}$ to the random oracle H, algorithm $\mathcal{B}$ searches L_{H} for a triple $(m, c, *)$. If it exists, $\mathcal{B}$ outputs c. Otherwise, $\mathcal{B}$ increments ℓ, queries its challenge oracle $c_\ell \leftarrow \mathsf{O}_{R_n}$, stores $(m_\ell \leftarrow m, c_\ell, \square)$ in L_{H}, and outputs c_ℓ. $\square$ serves as a placeholder for "uninitialized" and $*$ is a wildcard.

Signature queries. When $\mathcal{A}$ queries its signature oracle on m, algorithm $\mathcal{B}$ searches L_{H} for a triple (m_i, c_i, s_i). If it exists, $\mathcal{B}$ outputs s_i. Otherwise, $\mathcal{B}$ queries f_t^{-1} with $\mathsf{H}(m_i)$, receives s_i, stores (m_i, c_i, s_i) in L_{H}, and returns s_i to $\mathcal{A}$.

Output. When $\mathcal{A}$ stops, it outputs a forgery (m^*, σ^*). Assume $m^* = m_\jmath$. Let $L'_{\mathsf{H}} = \{(m^{(1)}, c^{(1)}, s^{(1)}), \ldots, (m^{(q_{\mathsf{I}})}, c^{(q_{\mathsf{I}})}, s^{(q_{\mathsf{I}})})\}$ be the set of all triples in L_{H}, excluding those of form $(*, *, \square)$. $\mathcal{B}$ sets

$$\pi = \{(i, j) : \exists a^{(i)} \in L'_{\mathsf{H}} \, \exists b_j \in L_{\mathsf{H}} : a^{(i)} = b_j\} \cup \{(q_{\mathsf{I}} + 1, \jmath)\}$$

and outputs $(\pi, s^{(1)}, \ldots, s^{(q_{\mathsf{I}})}, \sigma^*)$.

Analyis. Note that $\mathcal{B}$ perfectly simulates $\mathcal{A}$'s environment. Since H, and f_a are collision resistant, we can safely assume that $\mathcal{A}$ outputs a forgery on a message m^* that has never been sent to the signing oracle. Thus, $\mathcal{B}$ has not queried f_t^{-1} on $\mathsf{H}(m^*)$. Therefore, $\mathcal{B}$ makes $\imath = q_{\mathsf{I}}$ queries to f_t^{-1} and outputs $q_{\mathsf{I}} + 1$ preimages along with an injective map π. Thus, the first and last requirements in the CTTI experiment are met. As for the second requirement, we state that $f_a(s_i) = \mathsf{H}(m_{\pi(i)})$ for all $i \neq \jmath$ and $f_a(\sigma^*) = \mathsf{H}(m^*) = \mathsf{H}(m_{\pi(\jmath)})$. Therefore, $\mathcal{B}$ is successful whenever $\mathcal{A}$ is.

$\mathsf{CTTI} \Leftarrow \mathsf{GPV}$*:* Now, assume that the CTTI is not $(t, q_{\mathsf{I}}, q_{\mathsf{O}}, \epsilon)$-hard, i.e. there exists an adversary $\mathcal{A}$ that efficiently solves the problem. We show that $\mathcal{A}$ can be used to break strong unforgeability of GPV. We construct a forger $\mathcal{B}$ as follows.

Setup. $\mathcal{B}$ gets as input the public trapdoor parameter a. It sets up a list $L_{\mathsf{H}} \leftarrow \emptyset$ of triples (m, c, x), indexed by m. Furthermore, it initializes a counter $\ell \leftarrow 0$. It runs a black-box simulation of $\mathcal{A}$ on input a.

Random oracle H. On input m, $\mathcal{B}$ searches L_{H} for a triple $(m, c, *)$. If it exists, it outputs c. Otherwise, $\mathcal{B}$ increases ℓ, chooses a new $c_\ell \leftarrow R_n$, and stores $(m_\ell \leftarrow m, c_\ell, \square)$ in L_{H}, where $\square$ denotes "uninitialized" and * is a wildcard. Finally, $\mathcal{B}$ returns c_ℓ.

Challenge oracle queries. $\mathcal{B}$ chooses a new $m \leftarrow \{0,1\}^*$, computes $c \leftarrow \mathsf{H}(m)$, and returns c.

Inversion queries. On input c, $\mathcal{B}$ searches L_{H} for a triple (m_i, c, x_i). If it exists and $x_i \neq \square$ then $\mathcal{B}$ outputs x_i. If it does not exist then $\mathcal{B}$ increments ℓ, sets $i \leftarrow \ell$, chooses a new $m_i \leftarrow \{0,1\}^*$, and adds $(m_i, c, \square)$ to L_{H}. Finally, $\mathcal{B}$ queries $x_i \leftarrow f_t^{-1}(c)$, stores (m_i, c, x_i), and returns x_i.

Output. When $\mathcal{A}$ stops, it outputs $(\pi, x_1, \ldots, x_\jmath)$. Algorithm $\mathcal{B}$ searches the lowest index i, for which $(m_{\pi(i)}, c_{\pi(i)}, \square) \in L_{\mathsf{H}}$. It outputs the forgery $(m_{\pi(i)}, x_i)$.

Analysis. First of all, note that $\mathcal{B}$ perfectly simulates all of $\mathcal{A}$'s oracles. Since $\mathcal{A}$ is a successful chosen target trapdoor inverter, there is an index i with $f_a(x_i) = c_{\pi(i)}$, such that $\mathcal{A}$ never queried the inversion oracle on $c_{\pi(i)}$. Therefore, $\mathcal{B}$ has never queried its signature oracle on $\mathsf{H}(m_{\pi(i)}) = c_{\pi(i)}$ and x_i is a valid forgery on the message $m_{\pi(i)}$.

In both proofs, the number of inversion queries equals the number of signature queries and the number of challenge oracle oracle queries equals the number of queries to the random oracle. The overhead of handling $\mathcal{A}$'s queries is minimal and consists mainly of list operations that can be neglected because they are essentially the same in both reductions. This concludes the proof. □

Using the last theorem, we can now prove one-more unforgeability of our blind signature scheme.

Theorem 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 unforgebility of BS. Using $\mathcal{A}$, we construct an algorithm $\mathcal{B}$ via a black-block 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 an 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 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.

Again, the overhead of handling $\mathcal{A}$'s queries is dominated by simple list processing and can be neglected. □

Together with Theorem 2, our construction is one-more unforgeable if the GPV signature is strongly unforgeable.

4 Realization

The underlying signature scheme was developled by Gentry, Peikert, and Vaikuntanathan (GPV) and presented at STOC 2008 [13]. It uses a modified Babai nearest plane algorithm [4] and two famous results by Ajtai [2,3] in order to build a trapdoor function that is arguably "as good as" a trapdoor permutation. Its security is proven in the random oracle model and reduces to the collision resistance of f_a, which in turn reduces to the hardness of finding short vectors in a lattice. We refer to reader to [13] and to [17] for further details and a comprehensive discussion of the involved lattice problems and on Gaussians in the lattice context. The practical hardness of these lattice problems is analyzed in [5] and subsequently in [6].

GPV *trapdoor function.* The trapdoor function from [13] is defined as follows.

Public parameters. Depending on the security parameter n, the other parameters in [13] can be chosen as

$$\begin{array}{rl} \text{Modulus} & q = n^3\,, \\ \text{Domain dimension} & m = 5\,n\,\log(q)\,, \\ \text{Basis length bound} & L = m^{1+\epsilon}, \epsilon > 0\,, \\ \text{Gaussian parameter} & s = L\,\omega(\sqrt{\log(m)})\,. \end{array}$$

Spaces. The range is

$$R_n = \mathbb{Z}_q^n\,,$$

the (modified) domain D_n is

$$D_n = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\|_\infty \leq 2\,s\,\omega(\sqrt{\log(m)})\}\,,$$

and the range of the function SamplePre is

$$D_n^\star = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\|_\infty \leq s\,\omega(\sqrt{\log(m)})\}\,.$$

Trapdoor description. The public trapdoor key a describes the above public parameters and the public matrix

$$\mathbf{A} \in \mathbb{Z}_q^{n \times m}\,.$$

The set

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A}\,\mathbf{v} \equiv \mathbf{0} \pmod q\}$$

describes a lattice, for which the secret trapdoor paramter t describes a basis $\mathbf{T}$, such that

$$\left\|\tilde{\mathbf{T}}\right\| \leq L\,.$$

Here, $\tilde{\mathbf{T}}$ is the Gram-Schmidt orthogonalized matrix $\mathbf{T}$ and the norm of a matrix is defined as

$$\|\mathbf{X}\| = \left\| \begin{pmatrix} | & & | \\ \mathbf{x_1} & \cdots & \mathbf{x_c} \\ | & & | \end{pmatrix} \right\| = \max_{i=1,\ldots,c} \|\mathbf{x_i}\|_2\,.$$

Trapdoor evaluation. On input $\mathbf{x}$, the trapdoor function $f_a(\mathbf{x})$ evaluates to

$$\mathbf{y} \leftarrow \mathbf{A}\,\mathbf{x} \mod q.$$

Preimage sampling. Sampling from f_t^{-1} is performed via a modified Babai nearest plane algorithm. The algorithm explicitly uses $\mathbf{T}$ and relies on its short length. On input $\mathbf{y}$, it performs the following steps.

1. Compute $\mathbf{t} \in \mathbb{Z}_q^m$, such that $\mathbf{A}\,\mathbf{t} \equiv \mathbf{y} \pmod q$. This is done by linear algebra and most likely yields a $\mathbf{y} \notin D_n^\star$.
2. Use the trapdoor basis $\mathbf{T}$ to sample a vector $\mathbf{v}$ from a gaussian distribution around $-\mathbf{t}$ with standard deviation s and output $\mathbf{x} = \mathbf{t} + \mathbf{v}$

The described trapdoor function has all the properties mentioned in Section 3. As for the required linearity in our blind signature scheme, note that f_a is linear in the sense that for for all $\mathbf{x_1} + \mathbf{x_2} \in D_n$:

$$f_a(\mathbf{x_1} + \mathbf{x_2}) = f_a(\mathbf{x_1}) + f_a(\mathbf{x_2}) \mod q\,.$$

Therefore, all computations in D_n, $D_n^\star$, and R_n have to be performed modulo q.

Concerning security of the GPV signature scheme, we state that it is unforgeable if the problem of finding short integer solutions $\mathbf{v} \in \mathbb{Z}^m$, $\|\mathbf{v}\|_\infty \leq 2\,s\,\omega(\sqrt{\log(m)}$, of the equation

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

is hard [13]. As for our modified setting, with $D_n^\star$ and D_n, we need a slightly stronger assumption, i.e. the above problem has to be hard with $\|\mathbf{v}\|_\infty \leq 3\,s\,\omega(\sqrt{\log(m)}$. Furthermore, we claim that this special setting cannot be exploited to forge a signature $\sigma' \in D_n^\star$ from two valid signatures $\sigma_1, \sigma_2 \in D_n$ by simply adding them as $\sigma' \leftarrow \sigma_1 + \sigma_2$ because of the collision resistance of the full-domain hash H. We support intuition by a modified security proof for the GPV signature scheme.

Unforgeability of the Modified GPV *Signature* In the following, we adapt the proof from [13] (Proposition 6.1 in the extended version) to the modified signature scheme in Section 3.

Theorem 4. *Let $n, m, q, L, s, \Lambda^\perp(\mathbf{A})$ be as defined in Section 4 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 $(t, q_{\mathsf{Sig}}, \epsilon)$-strongly unforgeable if finding a vector $\mathbf{v} \in \Lambda^\perp(\mathbf{A})$ with*

$$\|\mathbf{v}\|_\infty \leq 3\,s\,\omega(\sqrt{\log(m)}$$

is (t', ϵ')-hard with

$$t' = t + q_{\mathsf{Sig}}\,(T_{\mathsf{SampleDom}}(n) + T_{f_a}(n)) \quad \textit{and} \quad \epsilon' = \epsilon - 2^{-\omega(\log(n))}\,.$$

Proof. Given a successful adversary $\mathcal{A}$ against strong unforgeabiliy with success probability ϵ, we build an algorithm $\mathcal{B}$ that finds a collision in f_a and, with that, a short vector in $\Lambda^\perp(\mathbf{A})$. Algorithm $\mathcal{B}$ runs $\mathcal{A}$ in a black-box simulation and works as follows.

Setup. Algorithm $\mathcal{B}$ receives the public parameter a of $\Lambda^\perp(\mathbf{A})$ as input sets up a list $L_{\mathsf{H}} \leftarrow \emptyset$ of triples (m, h, σ) in order to simulate H and f_a 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 $\sigma^* - \sigma$.

Analyis. 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 two three:

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 a valid signatures on m^*, we have

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

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

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

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

$$\|\sigma^* - \sigma\|_\infty \leq 2\, s\, \omega(\sqrt{\log(m)} + s\, \omega(\sqrt{\log(m)} \leq 3\, s\, \omega(\sqrt{\log(m)}\,.$$

The overhead of the reduction is dominated by the computational cost for domain sampling and trapdoor evaluation. Neglecting superfluous queries to H, there is one call to $\mathsf{SampleDom}$ and one call to f_a for each signature query of $\mathcal{A}$. □

Acknowledgements

The auther would like to thank Dominique Schröder for numerous fruitful discussions on the subject. Furthermore, the author thanks Michael Schneider and Richard Lindner for reviewing parts of this work.

References

1. 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.
2. 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.

3. Miklós Ajtai. *Generating Hard Instances of the Short Basis Problem.* ICALP 1999, Lecture Notes in Computer Science, pages 1–9. Springer-Verlag, 1999.
4. László Babai. *On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica*, 6(1):1–13, 1986.
5. Johannes Buchmann, Richard Lindner, and Markus Rückert. *Explicit hard instances of the shortest vector problem.* PQCrypto 2008, Lecture Notes in Computer Science, pages 79–94. Springer-Verlag, 2008.
6. Johannes Buchmann, Richard Lindner, and Markus Rückert. *Explicit hard instances of the shortest vector problem (extended version).* Number 2008/333 in Cryptology eprint archive. `eprint.iacr.org`, 2008.
7. 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.
8. 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.
9. 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.
10. Ran Canetti, Oded Goldreich, and Shai Halevi. *The random oracle methodology, revisited. J. ACM*, 51(4):557–594, 2004.
11. David Chaum. *Blind Signatures for Untraceable Payments.* Advances in Cryptology — Crypto 1982, pages 199–203. Plemum, New York, 1983.
12. 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.
13. 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, Lecture Notes in Computer Science, pages 197–206. Springer-Verlag, 2008.
14. 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.
15. 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.
16. Christoph Ludwig. *A Faster Lattice Reduction Method Using Quantum Search.* ISAAC 2003, Volume 2906 of Lecture Notes in Computer Science, pages 199–208. Springer-Verlag, 2003.
17. Daniele Micciancio and Oded Regev. *Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing*, 37(1):267–302, 2007.
18. 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.
19. 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.
20. David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology*, 13(3):361–396, 2000.
21. Oded Regev. *Quantum Computation and Lattice Problems.* Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2002, pages 520–529. IEEE Computer Society, 2002.
22. 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.