SNARK Friendly Crypto
Security Estimation, Hash and Signature

1 Notations

We denote with log the logarithm to base e, all other logarithms are specified, e.g., $\log_2$. Vectors and matrices are written in boldface, e.g., $\mathbf{v}$ and $\mathbf{M}$. We use $|\mathbf{v}|_p$ to denote the l_p norm of vector $\mathbf{v}$.

2 Preliminaries

Definition 1 (Lattice)**.** *A (full-dimensional) lattice in $\mathbb{R}^n$ is a discrete subgroup $L = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$, where typically $\mathbf{B} = [\mathbf{b_1}, \ldots, \mathbf{b_n}] \in \mathbb{Z}^{n\times n}$ is a matrix of linearly independent vectors. The matrix $\mathbf{B}$ is a basis of the lattice L and we write $L = L(\mathbf{B})$.*

Definition 2 (Determinant)**.** *Given any basis $\mathbf{B}$ of the lattice L, the determinant $det(L)$ of the lattice is $\sqrt{det(\mathbf{B}^\top\mathbf{B})}$. It is an invariant of the lattice.*

Definition 3 (Successive Minima)**.** *The i-th successive minimum, denoted as $L_i(L)$, is the smallest radius of a sphere that contains i linearly independent vectors in L.*

Definition 4 (Dual)**.** *For a lattice $L(\mathbf{B})$, its dual lattice is defined as $L^* = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}\cdot\mathbf{y} \in \mathbb{Z}, \forall y \in L(\mathbf{B})\}$.*

Definition 5 (SVP)**.** *Given a basis $\mathbf{B}$ of L and an approximation factor $\gamma \geq 1$, the task of SVP is to find a non-zero vector $v \in L$ with $|\mathbf{v}|_p \leq \gamma L_1$.*

Definition 6 (SIVP)**.** *Given a basis $\mathbf{B}$ of L and an approximation factor $\gamma \geq 1$, the task of SIVP is to find a set $\{\mathbf{v_1}, \ldots, \mathbf{v_n}\}$ of linearly independent vectors in L such that* $\max_i |\mathbf{v_i}|_p \leq \gamma L_n$

Remark. In [1], the authors reduce from SVP, SVIP in the l_2 norm to the corresponding problems in other norms, i.e., l_p norm where $1 \leq p \leq \infty$. That is to say, SVP, SIVP is hard regarding any l_p norm.

Definition 7 (q-ary)**.** *A lattice L is called a q-ary if $q\mathbb{Z} \subseteq L$.*

For $q \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define the two most important q-arys (SIS is defined upon one of them).

$$\Lambda_q(\mathbf{A}) = \{\mathbf{w} \in \mathbb{Z}^n \mid \exists \mathbf{e} \in \mathbb{Z}^m \mathbf{A}^\top \mathbf{e} = \mathbf{w}\,(mod\ q)\} \tag{1}$$

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{v} = \mathbf{0}\,(mod\ q)\} \tag{2}$$

Proposition 1. *$\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are mutual dual lattice by a scaling factor. Specifically, $\Lambda_q^\perp(\mathbf{A}) = q\Lambda_q(\mathbf{A})^*$ and $q\Lambda_q^\perp(\mathbf{A})^* = \Lambda_q(\mathbf{A})$.*

Proposition 2. *Let q be a prime and $m = O(n\log(n))$. With high probability, the rows of $\mathbf{A}$ are linearly independent over $\mathbb{Z}_q$ and $det(\Lambda_q^\perp(\mathbf{A})) = q^n$.*

Definition 8 (SIS)**.** *Given $n, m, q \in \mathbb{N}$, a randomly picked $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a norm bound $1 \leq \beta < q$, SIS problem is to find $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$ with $0 < |\mathbf{v}|_p \leq \beta$.*

Remark. There is a famous worst-case to average-case reduction from SIVP to SIS.

Definition 9 (LWE)**.** *Given $n, m, q, \alpha \in \mathbb{N}$, a randomly picked $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and $\mathbf{b} \in \mathbb{Z}_q^m$, LWE problem is to recover $\mathbf{s}$ such that $\mathbf{b} = \mathbf{A}^\top\mathbf{s} + \mathbf{e}$ $(mod\ q)$, where $\mathbf{e}$ is chosen from the discretized normal distribution with 0 mean and standard deviation of $\alpha q/2\pi$.*

Remark. There is a famous worst-case to average-case reduction from BDD to LWE.

3 Estimating The Security

Almost every lattice-based crypto is built upon the hardness of SIS or LWE. We discuss how to estimate the level of security of SIS and LWE. This is a crucial step as it provides a methodological way to configure the parameters of different types of crypto system with high level security guarantee.

In practice, it is unlikely that λ_1 is known, we can only estimate it. A natural heuristic is to estimate $\lambda_1(L)$ as the smallest radius of a ball whose volume is $det(L)$ (The volume of L's fundamental region) [2], which is roughly proportional to $det(L)^{1/dim(L)}$. For q-ary $\Lambda_q^\perp(\mathbf{A})$ in particular, Gama and Nguyen find the shortest non-zero vector found by the best known algorithm is close to

$$\min\{q, (det(\Lambda_q^{\perp}(\mathbf{A})))^{1/m}\delta^m\} = \min\{q, q^{n/m}\delta^m\}, \tag{3}$$

where the equality holds with high probability when q is prime and m is not close to n [3]. They find that when the dimension is not too small > 500, δ is a parameter with dominant role. Regardless of other parameters, there is no known algorithm can achieve $\delta < 1.011$, and they therefore predict that $\delta = 1.005$ is totally out of reach [3].

Definition 10 (Hermite-SVP)**.** *Given a $d \times d$ (full dimensional) lattice $L = L(B)$ and a approximation factor δ, δ-HSVP is to find a short vector $\mathbf{v} \in L$ such that $|\mathbf{v}|_p \leq \delta^d det(L)^{1/d}$. In particular, for q-ary $\Lambda_q^{\perp}(\mathbf{A})$ with $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, δ-HSVP is to find $\mathbf{v}$ such that $|\mathbf{v}|_p \leq \delta^m q^{n/m}$.*

In [4], they have conducted a series of experiments that runs δ-HSVP on randomly generated $\Lambda_q^{\perp}(\mathbf{A})$. Their experiments use l_2 norm, $n > 128$ (the number of rows in A) and $q = n^3$. Based on their experimental results, they have the following conjecture.

Conjecture 1. *Let $n > 128$, $m = O(n \log n)$ and $q > n^2$. The shortest vector can be found by the best known algorithm in $\Lambda_q^{\perp}(\mathbf{A})$ has the form of the $\delta-$HSVP, i.e., $q^{n/m}\delta^m$. Morevoer, HSVP on $\Lambda_q^{\perp}(\mathbf{A})$ is dominantly affected by the approximation factor δ.*

In our case, q is a very large prime. We should achieve similar level of security on δ-HSVP with significantly smaller n (e.g. n= 4). We'll need some experimental results to support this.

Proposition 3. *Let q be a large enough prime, $m = O(n \log n)$ and $\beta < q$. Let $SIS(n, m, q, \beta)$ be the problem of finding a short vector $\mathbf{v}$ in q-ary $\Lambda_q^{\perp}(\mathbf{A})$ with $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, such that $|\mathbf{v}| \leq \beta$. Let matrix $\mathbf{A}' \in \mathbb{Z}_\mathbf{q}^{\mathbf{n}\times\mathbf{d}}$ be the matrix by removing the last $m - d$ colummns from $\mathbf{A}$. The optimum attack to $SIS(n, m, q, \beta)$ by a HSVP solver S is to run S on $\Lambda_q^{\perp}(\mathbf{A}')$ with $\delta \leq (\beta/q^{n/d})^{1/d}$, where $d = \min\{x \in \mathbb{N} \mid q^{2n/x} \leq \beta\}$.*

Proof. The proof consists of two parts.

1. First, if $\mathbf{v}' \in \Lambda_q^{\perp}(\mathbf{A}')$, then $\mathbf{v}$, which is obtained by appending $m - d$ zeros to v', is in $\Lambda_q^{\perp}(\mathbf{A})$. Obviously, $|\mathbf{v}'|_p = |\mathbf{v}|_p$.

2. Removing $m-d$ colummns from $\mathbf{A}$ gets us a new matrix $\mathbf{A}'$. It is pretty easy to show $\Lambda_q^{\perp}(\mathbf{A}')$ still has determinant q^n with high probability.

year	Standard (2018)	2010	2020	2030	2040	2050	2060	2070	2080	2090	2100
bit security	SHA/AES	75	82	88	95	102	108	115	122	128	135
λ	160	225	246	264	285	306	324	345	366	384	405
κ	128	150	164	176	190	204	216	230	244	256	270
Hacker	1.00993	1.01177	1.00965	1.00808	1.00702	1.00621	1.00552	1.00501	1.00458	1.00419	1.00389
Lenstra	1.00803	1.00919	1.00785	1.00678	1.00602	1.00541	1.00488	1.00447	1.00413	1.00381	1.00356
Int. agency	1.00710	1.00799	1.00695	1.00610	1.00548	1.00497	1.00452	1.00417	1.00387	1.00359	1.00336

Figure 1: Infeasible parameters for HSVP [4]. The upper rows present recommended post- quantum secure symmetric key size $\|$ and hash function length. Each of the lower cells contains an upper bound for the HSVP-parameter, such that this problem is computationally hard for the given attacker (row) until the end of a given year (column).

3. Given d, the function $\delta^d q^{n/d}$ obtained minimum when $\delta = 2^{n \log_2 q/d^2}$. In a consequence, a sufficiently good HSVP solver in dimension d should be able to find vector of length $2^{n \log_2 q/d} q^{n/d} = q^{2n/d}$. We need make sure $q^{2n/d} \leq \beta$ and hence S works for $\delta \leq (\beta/q^{n/d})^{1/d}$ ■

We want to conduct experiments and show that the condition $n > 128$ is not necessary. It should be the case whenever the optimal dimension d is large enough (e.g., $d > 256$). Also, we want to show the choice of norm is irrelevant.

Proposition 4. *Given $SIS(n, m, q, \beta)$, we can evaluate its level of security by the following algorithm.*

λ	cost	output length	n	m	attacking dimension	delta
82	2384	596	2	1192	234	1.0076
102	9536	1192	4	2384	425	1.0046

Table 1: Minimum Parameter Configuration for Ajtai Hash for different security level λ. The prime model is set to q_4 in [5].

1. *Check if $m \geq 2n \log n$, $q \geq n^2$ (unnecessary in SNARK), $\beta < q$ (unnecessary in SNARK)*
2. *Compute $d = \lceil 2n \log q / \log \beta \rceil$. Check if $d > 256$.*
3. *Compute $\delta = (\beta / q^{n/d})^{1/d}$. Check its corresponding level of security based on tables in [4].*

Proposition 5 (from LWE to SIS)**.** *A successful adversary against $SIS(n, m, q, 1.5\sqrt{2\pi}/\alpha)$ can also break $LWE(n, m, q, \alpha)$.*

4 Hash Functions

Definition 11 (Ajtai Hash)**.** *Given $n, m, q \in \mathbb{N}$, a randomly picked $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the Ajtai Hash $h : \{0,1\}^m \to \mathbb{Z}_q^n$ is defined as*

$$h(\mathbf{x}) = \mathbf{A}\mathbf{x} \ (mod\ q) \tag{4}$$

Suppose an adversary can find a collision for $Ajtai(m, n, q)$, i.e., $\mathbf{x_1} \neq \mathbf{x_2}$ and $h(\mathbf{x_1}) = h(\mathbf{x_2})$. Then, $h(\mathbf{x_\Delta}) = h(\mathbf{x_1} - x_2) = \mathbf{0}$, where $\mathbf{x_\Delta} \in \{-1, 0, 1\}^m$ and therefore he successfully solves $SIS(m, n, q, \sqrt{m})$ under l_2 norm.

We use $SIS(m, n, q, \sqrt{m})$ to estimate the level of security of $Ajtai(m, n, q)$. In particular, we run the algorithm in Proposition 4, with $q = q_4$ where q_4 is the large prime used in [5], to find the minimum n (therefore minimum m) such that its corresponding HSVP problem has $\delta < 1.005$. Notice that the parameters in Table 2 are the minimum configuration. Larger m (the bit-wise length of input), for example, is allowed and will only increase the level of security.

Definition 12 (GCK Hash)**.** *Given $n, m, q \in \mathbb{N}$, a ring $R = \mathbb{Z}_q / \langle x^n + 1 \rangle$, a randomly picked $\mathbf{A} \in R^m$, the GCK Hash $h : D^m \to R$ is defined as*

$$h(\mathbf{x}) = \Sigma_i \mathbf{A_i} \cdot \mathbf{X_i} \ (mod\ q), \tag{5}$$

λ	cost	output length	n	m	d_D	attacking dimension	delta
82	928	596	4	116	3	330	1.0076
102	1856	1192	8	116	3	617	1.0045

Table 2: Minimum Parameter Configuration for GCK Hash for different security level λ. The prime model is set to q_4 in [5].

where each $\mathbf{A_i}$, $\mathbf{X_i}$ *is a ring in* R, $\cdot$ *denote the polynomial product within* R, *and* $D = \{\mathbf{y} \in R \,|\, |\mathbf{y}|_\infty \leq d_D\}$ *for some* $d_D > 0$.

To make a GCK Hash collision-resistant, the following two conditions need to be met. See [8] for detailed proof.

1. $m > \log q / log 2 d_D$
2. $p > 4dmn^{1.5} \log n$

In terms of security level, it is shown that $SIS(n, mn, q, 2\sqrt{mn}d_D)$ can be reduced to $GCK(n, m, q, d_D)$, as polynomial production of two rings $\mathbf{a} \cdot \mathbf{x} \in R$ can be represented by the product of the skew-circulant matrix presentation of $\mathbf{a}$ and vector $\mathbf{x}$ [9].

we run the algorithm in Proposition 4 to find the minimum $m * n$ such that $\delta < 1.005$. See Table **??**.

Computation cost Analysis: Problem & Possible Solution. In SNARK system, the computation cost depends on the number of multiplication required. Take Ajtai hash as an example, the computation cost consists of two parts.

1. mn multiplication for computing $\mathbf{Ax}$, which means 9536 multiplication gates under minimum parameter setting.
2. Checking input $\mathbf{x}$ is indeed binary.

(Thanks to Ahemd for explaining this.) For the second part of cost, it will need 1 multiplication gate to check 1 bit. So in total it needs m extra multiplcation. However, it turns out that the real computation cost for the multiplication of the two types differs a lot. This is because (1) in the first type the multiplication is for a variable and a constant (2) while in the second type the multiplication is between two variables. **Unfortunately, the cost for the second type of multiplication is so big that m such multiplication is already too much.**

Proposition 6. *For any vector* $\mathbf{x}$ *of size* m*, checking that* $|\mathbf{x}|_\infty \le b$ *requires* $m * \log_2(b)$ *(the bit-wise length of* $\mathbf{x}$*) multiplication of two variables.*

Bad news. Crypto based on SIS will always has some norm bound checking. I also think the majority, if not all, of lattice-based crypto have this issue.

Possible Solution. Recall that SVP and SIVP is of the same hardness in different norm $l_p, 1 \le p \le \infty$ [1]. Consider the following l_1-Ajtai hash function.

Definition 13 (l_1-Ajtai Hash)**.** *Given* $n, m, q \in \mathbb{N}$*, a randomly picked* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*, the Ajtai Hash* $h : D \to \mathbb{Z}_q^n$ *is defined as*

$$h(\mathbf{x}) = \mathbf{A}\mathbf{x}\ (mod\ q), \tag{6}$$

where $D \subseteq \mathbb{Z}_q^m$ *and* $\forall \mathbf{x} \in D, |\mathbf{x}|_1 \le m$*.*

Following a similar proof, it is easy to see if an adversary can find a collision for l_1-Ajtai(n,m,q), then he can solve l_1-SIS$(n, m, q, 2m)$. So we can estimate the level of security of l_1-Ajtai(n,m,q) based on l_1-SIS$(n, m, q, 2m)$. We can run experiments to build the security level table for l_1-SIS.

Proposition 7. *Assume we can efficiently check if a summation exceeds the module* q *in SNARK. For any vector* $\mathbf{x}$ *of size* m*, checking* $|\mathbf{x}|_1 \le b$ *requires* $\log_2 b$ *multiplication of two variables.*

Ahemd thinks there is no straightforward way to check if a summation exceeds the module q in SNARK. If so, we have a big problem on SIS-based crypto, because none of norms can be checked efficiently.

5 Signature without Trapdoors

We focus on the signatures proposed in [7].

SIS-Based Signature.

See Figure 2. The key idea on its security against chosen plaintext attack is to show the two signing algorithm in Figure 3 are statistically close.

We list the conditions on the parameters.

1. For the random oracle, $2^\kappa \binom{k}{\kappa} \ge 2^100$. In our case, we (probably) use SHA-256 for the random oracle, whose output has 256 binary bits and hence $k = \kappa = 256$.

2. For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a uniformly picked $\mathbf{s}$ from $\{-d, \ldots, 0, \ldots, d\}^m$, we need to guarantee that with probability at least $1 - 2^{-100}$ there exists a different $s' \in \{-d, \ldots, 0, \ldots, d\}^m$ such that $\mathbf{As} = \mathbf{As}'$.
$$m > (n \log q + 100)/\log(2d + 1) \tag{7}$$
3. For the signature $\mathbf{z}$, we need $|\mathbf{z}|_2 \leq \eta\sigma\sqrt{m}$ with probability $> 1-2^{-100}$.
$$\eta e^{(1-\eta^2)/2} < 2^{-100/m} \tag{8}$$
4. If an adversary under CPA model can attack the signature in Figure 2, then he can solve l_2-$SIS(n, m, q, \beta)$, where $\beta = 2(\eta\sigma + dk)\sqrt{m}$.

We calculate the computation cost of signing/verifying algorithm.

Signing Algorithm

1. (Line 1) Generating a sampel $\mathbf{y}$ from discrete Gaussian distribution D_σ^m. Use a PRF(SHA256) to generate a uniformly random seed. Then, compute Box?Muller transform, or use Ziggurat algorithm.
2. (Line 2) Call the random oracle H. The problem is the input length of H here is $n\lceil \log q\rceil$+(message bit-wise length), where $\lceil \log q\rceil = 298$. If we use SHA-256 to construct H, we will need at least $\lceil n * 298/512\rceil$ SHA-256 gadgets.
3. (Line 3) Need mk multiplications.
4. (Line 4) It outputs with probability $\approx 1/M$. Hence, the expected cost of the signing algorithm is M times the cost from the first three lines.

Verifying Algorithm

1. Verify the l_2 norm of $\mathbf{z}$, which costs approximately (?) $m \log(\eta\sigma\sqrt{m})$ multiplication.
2. Verify that $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc}, \mu)$, which needs $nm + nk$ multiplications plus the cost of calling the random oracle.

LWE-Based Signature.

The signature in Figure 2 can be modified by letting the secret key $\mathbf{s}$ sampled from D_ψ^{2n}, where $\psi = \sqrt{d(d+1)/3}$. The observation is if we have $\mathbf{A} = [\bar{\mathbf{A}}|\mathbf{I}] \in \mathbb{Z}_q^{n\times 2n}$ with $\bar{\mathbf{A}}$ randomly chosen from $\mathbb{Z}_q^{n\times n}$. Then, distinguishing pairs $(\mathbf{A}, \mathbf{As})$, where $\mathbf{s}$ sampled from D_ψ^{2n}, from uniformly distributed pairs in $\mathbb{Z}_q^{n\times 2n} \times \mathbb{Z}_q^{n\times n}$ is exactly the decisional LWE problem. By setting $\psi = \sqrt{d(d+1)/3}$, the secret keys generated by the two different approaches will have approximately the same length. As a consequence, condition (2) in the SIS-based signature is not necessary anymore.

λ	sign cost	verify cost	n	m	d	η	M	SHA-256 Gadgets	δ
82	1251413	328084	86	2572	512	1.2	1.824	33	1.0078
102	1863710	670092	130	3883	512	1.3	1.824	49	1.0053

Table 3: Minimum Parameter Configuration for Lyu12-SIS Signature. The cost computation only considers using 1 SHA-256 Gadget, which is approximately 27500.

λ	sign cost	verify cost	n	m	d	η	M	SHA-256 Gadgets	δ
82	80061	11520	32	64	1	2.2	1.824	19	1.0076
102	95007	18816	48	96	1	2.0	1.824	28	1.0053

Table 4: Minimum Parameter Configuration for Lyu12-LWE Signature. The cost computation only considers using 1 SHA-256 Gadget, which is approximately 27500.

6 References

1. http://www.cims.nyu.edu/ regev/papers/lpnorm.pdf
2. https://www.cims.nyu.edu/ regev/papers/pqc.pdf
3. ftp://ftp.di.ens.fr/pub/users/pnguyen/Euro08.pdf
4. https://eprint.iacr.org/2010/137.pdf
5. https://eprint.iacr.org/2014/595
6. https://eprint.iacr.org/2011/537
7. http://www.di.ens.fr/ lyubash/papers/FSAbortAsiacryptconf.pdf
8. http://www.di.ens.fr/ lyubash/papers/generalknapsackfull.pdf
9. http://www.eecs.harvard.edu/ alon/PAPERS/lattices/swifft.pdf

Signing Key: $\mathbf{S} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$
Verification Key: $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{T} \leftarrow \mathbf{AS}$
Random Oracle: $\mathrm{H} : \{0,1\}^* \rightarrow \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$

Sign$(\mu, \mathbf{A}, \mathbf{S})$
1: $\mathbf{y} \xleftarrow{\$} D_\sigma^m$
2: $\mathbf{c} \leftarrow \mathrm{H}(\mathbf{Ay}, \mu)$
3: $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$
4: output $(\mathbf{z}, \mathbf{c})$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{M D_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$

Verify$(\mu, \mathbf{z}, \mathbf{c}, \mathbf{A}, \mathbf{T})$
1: Accept iff
$\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ and $\mathbf{c} = \mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu)$

Figure 2: Signature

Hybrid 1

Sign($\mu, \mathbf{A}, \mathbf{S}$)

1: $\mathbf{y} \xleftarrow{\$} D_\sigma^m$
2: $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
3: $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}$
4: with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{M D_{\mathbf{Sc},\sigma}^m(\mathbf{z})}, 1\right)$,
5: output $(\mathbf{z}, \mathbf{c})$
6: Program $\mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$

Hybrid 2

Sign($\mu, \mathbf{A}, \mathbf{S}$)

1: $\mathbf{c} \xleftarrow{\$} \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$
2: $\mathbf{z} \xleftarrow{\$} D_\sigma^m$
3: with probability $1/M$,
4: output $(\mathbf{z}, \mathbf{c})$
5: Program $\mathrm{H}(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$

Figure 3: Hybrid Signatures