NSUCRYPTO23 Problems

October 23, 2023

Problem 6.

The scheme has five algorithms: **KeyGen**, **Sign**, **AggregateSign**, **AggregateVerify**. In this problem, we consider a modified version the lattice signature scheme from Fiat-Shamir with abort due to Lyubachevsky et al. [1, 2] to define such a scheme. More specifically, we define the three algorithms **KeyGen**, **Sign**, **AggregateVerify** based from this modified version. Now let B and γ be bound parameters. Let $R = \mathbb{Z}[\mathbb{X}]/\langle X^n + 1\rangle$, where n is a power of 2. Consider a prime $q \equiv 1 \pmod{2n}$ and let $R_q = R/qR$. Denote $S = \{x \in R_q \mid ||x|| \leq \gamma\}$. Let $\mathbf{A} \in R_q^{k\times\ell}$ to be a public matrix. Now, the scheme proceed as follows.

KeyGen(1^λ): This algorithm is used to setup the parameters for all participants.

1. Sample $\mathbf{s}_i \in S^\ell$ for each $i = 1, 2, \ldots, n$ and let $sk_i = \mathbf{s}, pk_i = \mathbf{As} \pmod{q}$.
2. Return $(sk_i, pk_i)_{i=1}^n$.

Sign(sk_i, m): This algorithm is used by each signer to create a partial signature s_i of a message m. It proceed as follows:

1. Sample $\mathbf{y} \leftarrow S^\ell$.
2. Compute $\mathbf{t} = \mathbf{Ay} \pmod{q}$.
3. Compute $c = \mathsf{H}(m, \mathbf{t})$.
4. Compute $\mathbf{z} = \mathbf{y} + c \cdot sk_i \pmod{q}$
5. If $||z|| > B$ then restart from step 1, otherwise output $s_i = (\mathbf{t}, \mathbf{z})$.

AggregateSign$(m, s_1 = (\mathbf{t}_1, \mathbf{z}_1), \ldots, s_n = (\mathbf{t}_n, \mathbf{z}_n))$ This algorithm is used to aggregate all the signatures of n participants. It proceed as follows:

1. Compute $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_n$.
2. Compute $\mathbf{z} = \mathbf{z}_1 + \mathbf{z}_2 + \cdots + \mathbf{z}_n$.

3. Return $s = (\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n \mathbf{t}, \mathbf{z})$.

AggregateVerify$(m, pk, pk_1, pk_2, \ldots, pk_n, s = (\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n, \mathbf{t}, \mathbf{z}))$: This algorithm is used to verify the validity the aggregated signature of m. It proceed as follows:

1. For each $i = 1, 2, \ldots, 1000$ compute $c_i = \mathsf{H}(m, \mathbf{t}_i)$
2. If $\mathbf{Az} = c_1 \cdot pk_1 + c_2 \cdot pk_2 + \cdots + c_n \cdot pk_n + \mathbf{t}$ and $||\mathbf{z}|| \leq nB$ and $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_n$ then return 1, otherwise return 0.

Compared to the orginial version, instead of returning c_i, we let the signer returns $\mathbf{t}_i$ and the verifier can compute c_i from $\mathbf{t}_i$ so that the signature can be aggregated. We note that given $\mathbf{t}_i$, it is hard to find $\mathbf{y}_i$ s.t $\mathbf{Ay}_i = \mathbf{t}_i$ due to the Short Integer Solution assumption, hence $\mathbf{y}_i$ and sk_i remains safe.

References

[1] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In E. Prouff and P. Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[2] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.