Papercraft: Lattice-based Verifiable Delay Function

Aalto University
School of Science

Michał Osadnik[1], Darya Kaviani[2], Valerio Cini[3], Russell W. F. Lai[1], Giulio Malavolta[3]
[1]Aalto University, Finland [2]UC Berkeley, USA [3]Bocconi University, Italy

Introduction

Timed cryptography studies a family of cryptographic primitives with diverse functionalities designed to meet their security goals only for a short (polynomial) amount of time. This includes, for example, time-lock puzzles, timed-commitments, proofs of sequential work, verifiable delay functions, and delay encryption.

Verifiable Delay Functions (VDFs) let a prover show they spent a specific amount of time running a function, and the verifier can quickly check the result.

A VDF protocol includes these steps:

- Setup: Creates public parameters.
- GenInput: Generates the input for the function.
- Execute: Runs the function sequentially to get the output.
- Prove: Creates a proof showing the function was executed correctly.
- Verify: Checks the proof to confirm the execution.

From the efficiency perspective, Execute should not be computable in a time $< T$.

Verifiable Delay Function is traditionally on the following ingredients:

(a) sequential function,

(b) commitment to the trace of the computation of the sequential function,

(c) argument system for proving the correctness of the committed trace of the computation.

Pre-quantum: Repeated Squaring

The state of timed cryptography in pre-quantum settings is largely unsatisfactory. Virtually all efficient schemes are based on the hardness of a single problem (or variants thereof), namely the sequential squaring assumption. Loosely speaking, such an assumption postulates that the repeated application of the function

$$f(x) = x^2 \bmod N,$$

where $N = pq$ is an RSA modulus, is the fastest algorithm to compute $x^{2^T} \bmod N$ given x. In other words, there is no better algorithm than T-sequential iterations of f, provided that the order of the group is unknown by the evaluator.

Post-quantum assumption

In [1], a new candidate family of sequential functions is put forward, which is closely connected with lattice-based cryptography. Specifically, the new sequential function is defined to be the T repeated application of the binary decomposition operation followed by a SIS-based collision-resistant hash function [2], designed so that the domain and range of the function are the same.

Specifically, the base function $f_{\mathbf{A}} : \mathcal{R}_q^n \to \mathcal{R}_q^n$ is defined as:

$$f_{\mathbf{A}}(\mathbf{z}) \coloneqq -\mathbf{A}\mathbf{G}^{-1}(\mathbf{z}) \bmod q.$$

Here, $\mathcal{R}$ is a subring of a cyclotomic ring. Other PQ-secure candidates, such as MinRoot and ZKBdf, rely on different sequentiality assumptions but have not demonstrated the same level of practicality as our solution.

vSIS Assumption & Commitments

Short Integer Solution (SIS) is a well-known lattice problem that is used as a building block for many cryptographic primitives. The SIS problem (in ring variant) is defined as follows: given a random matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, a target vector $\mathbf{t} \in \mathcal{R}_q^n$, and a bound β, find a short (bounded by β) vector $\mathbf{x} \in \mathcal{R}_q^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{t} \bmod q$. SIS-bases commitment of a (short) witness $\mathbf{w} \in \mathcal{R}_q^m$ is $\mathbf{c} \in \mathcal{R}_q^n$ such that

$$\mathbf{A}\mathbf{w} = \mathbf{c} \bmod q,$$

where $\mathbf{A}$ is a random matrix and $\mathbf{w}$ is the witness. The commitment scheme is binding if it is hard to find a different witness $\mathbf{w}'$ such that $\mathbf{A}\mathbf{w}' = \mathbf{c}$ under the SIS assumption.

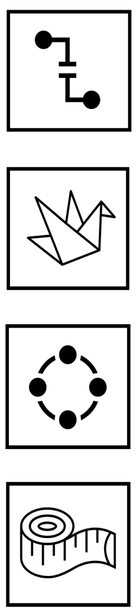
Vanishing SIS (vSIS) is a new assumption [3] which is similar to the SIS assumption, but $\mathbf{A}$ yields a row-tensor structure, i.e. $\mathbf{A} = \mathbf{A}_0 \bullet \ldots \bullet \mathbf{A}_{\mu-1}$, where $\mathbf{A}_i \in \mathcal{R}_q^{n \times d}$ for $i \in [\mu]$ and $\bullet$ denotes the row-wise Kronecker product (also called face-splitting product).

vSIS-based Argument System

The heart of Papercraft is a succinct argument system of [4], extending the work of [3], that proves the knowledge of (short) solution to the linear relations over the ring $\mathcal{R}_q$, i.e. knowledge of a witness $\mathbf{w} \in \mathcal{R}_q$ such that:

$$\mathbf{F}\mathbf{w} = \mathbf{y} \bmod q \quad \text{and} \quad \|\mathbf{w}\|_{\sigma,2} \leq \beta$$

where the matrix $\mathbf{F}$ has a row-tensor structure. The argument system of [4] consists of the following main ingredients:



Π^{split} and Π^{fold} designed to reduce the dimension of the witness at the expense of increased norm,

$\Pi^{b\text{-}\mathsf{decomp}}$ to decrease the norm (eliminating the correctness gap) at the expense of expanding the number of columns of the witness,

Π^{norm} to serve as an intermediate opening to the norm, effectively proving that the norm at the given point is exactly β. This protocol therefore "upgrades" the norm during the extraction.

Furthermore, [4] has been equipped with Π^{bin}, which is conceptually similar to the Π^{norm} protocol, but it is designed to prove that the witness is a bit string, i.e. $\mathbf{w} \in \mathcal{R}_2^m$. As a consequence, a combination of these protocols yields a succinct argument system for proving the the linear relation with exact coeficcient norm.

About Speaker

I am a third-year Ph.D. student. My research primarily focuses on lattice-based argument systems, with interests spanning theoretical advancements as well as the practical efficiency of implemented protocols. Beyond argument systems, I also explore areas such as time-based cryptography and fully homomorphic encryption (FHE).

I am looking for internships and collaborations. If you are interested in collaborating or have any questions, please feel free to reach out to me at `michal.osadnik@aalto.fi`.

Combining Argument System and Sequential Function

The sequential function discussed earlier is a linear relation over the ring $\mathcal{R}_q$. The trace of this function's computation is presented as witness $\mathbf{w}$ to a staircase-shaped matrix $\mathbf{F}$, which is a combination of the binary gadget matrix $\mathbf{G}$ and the random matrix $\mathbf{A}$. The relation can be expressed as:

$$\begin{pmatrix} \mathbf{G} & & & & \\ \mathbf{A} & \mathbf{G} & & & \\ & \mathbf{A} & \mathbf{G} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{A} & \mathbf{G} \\ & & & & \mathbf{A} \\ & & & \mathsf{ck} & \end{pmatrix} \mathbf{w} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{y}_T \\ \mathbf{c} \end{pmatrix},$$

where the appended bottom-most row $\mathbf{c}$ is a commitment to the witness $\mathbf{w}$, and the first and last vectors are the input and output of the function, respectively. The matrix $\mathbf{G}$ is a binary gadget matrix, while $\mathbf{A}$ is a random matrix.

Staircase Relation

The aforementioned relation of the VDF induces a *staircase-shaped matrix* formed by interleaving binary gadget matrix $\mathbf{G}$ and the hash matrix $\mathbf{A}$.

However, the staircase matrix is not a row-tensor structure and thus not immediately suitable for the aforementioned argument system. We introduce $\Pi^{\mathsf{staircase}}$, a protocol that transforms the staircase matrix into a row-tensor. The structure is imposed by "batching", i.e. left-multiplying, both sides of the staircase relation with $\mathbf{c}^{\mathrm{T}} := (c^0, \ldots, c^{n-1})$, a challenge vector with random c sent by the verifier.

Concrete Evaluation

We implemented Papercraft in Rust and conducted experiments on a high-performance computing node.[a] Our implementation comprises $\approx$ 7000 lines of Rust code, complemented by SageMath scripts. Parallelism was extensively employed, facilitated by the Rayon library. For core arithmetic, we optimised multiplication in small cyclotomic rings using the Karatsuba algorithm, while employing number theoretic transform for larger polynomial convolutions. In our evaluations, we measured prover runtime, verifier runtime, proof size, and scaling behaviour across different security levels and delay parameters. The most favourable experiments showed Papercraft achieves verification in 7 seconds for VDF computations requiring over 6 minutes. Proof sizes were $\approx$ 15 MB, while the witness size was 80MB. The prover runtime was $\approx$ 4 hours.

Code & Paper

References

[1] R. W. F. Lai and G. Malavolta, "Lattice-based timed cryptography," in *Crypto*, 2023.

[2] M. Ajtai, "Generating hard instances of lattice problems," in *STOC*, 1996.

[3] V. Cini, R. W. F. Lai, and G. Malavolta, "Lattice-based succinct arguments from vanishing polynomials," in *Crypto*, 2023.

[4] M. Klooß, R. W. F. Lai, N. K. Nguyen, and M. Osadnik, "RoK, paper, SISsors," in *Asiacrypt*, 2024.

[a]The calculations presented above were performed using computer resources within the Aalto University School of Science "Science-IT" project.