Lattice-Based Timed Cryptography

Giulio Malavolta
(Max Planck Institute —> Bocconi University)

Joint work with Russell Lai
Timed Cryptography

0 → Solution!

T
Timed Cryptography

- Time-Lock Puzzles
Timed Cryptography

- Time-Lock Puzzles
- Proofs of Sequential Work
Timed Cryptography

- Time-Lock Puzzles
- Proofs of Sequential Work
- Verifiable Delay Functions
Applications

- Seal-Bid Auctions
- E-Voting
- Randomness Generation
- Contract Signing
More Applications

chia

ethereum
Hardness vs Fine-Grained Hardness
Hardness vs Fine-Grained Hardness

One-Way Problems

\[ \mathsf{P} \neq \mathsf{NP} \]

\{ poly-size \}
Hardness vs Fine-Grained Hardness

One-Way Problems

\[ P \neq NP \]

Sequential Problems

\[ \mathsf{NC} \neq P \]

\[ \mathsf{P} \neq \mathsf{NP} \]
Landscape of Sequential Problems
Landscape of Sequential Problems

Repeated Squaring

\[ f_N(x) = x^2 \mod N \]
Landscape of Sequential Problems

Repeated Squaring

\[ f_N(x) = x^2 \mod N \]

Random Oracles

\[ f_\mathcal{H}(x) = \mathcal{H}(x) \]
Landscape of Sequential Problems

Structured

Repeated Squaring
\[ f_N(x) = x^2 \mod N \]

Isogeny Shortcut
\[ f_\Phi(x) = \Phi(x) \]

Unstructured

Random Oracles
\[ f_\mathcal{H}(x) = \mathcal{H}(x) \]

Real-World

Theory-World
Landscape of Sequential Problems

- **Repeated Squaring**
  \[ f_N(x) = x^2 \mod N \]

- **Isogeny Shortcut**
  \[ f_\Phi(x) = \Phi(x) \]

- **Random Oracles**
  \[ f_\mathcal{H}(x) = \mathcal{H}(x) \]

- **Universal Functions**
  Obfuscation, FHE

---

<table>
<thead>
<tr>
<th>Structured</th>
<th>Unstructured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-World</td>
<td>Theory-World</td>
</tr>
</tbody>
</table>
Enter Quantum Computing

Repeated Squaring
\[ f_N(x) = x^2 \mod N \]

Isogeny Shortcut
\[ f_\Phi(x) = \Phi(x) \]

Random Oracles
\[ f_\mathcal{H}(x) = \mathcal{H}(x) \]

Universal Functions
Obfuscation, FHE

Structured

Unstructured

Real-World

Theory-World
Enter Quantum Computing

Structured

Repeated Squaring
\[ f_N(x) = x^2 \mod N \]

Random Oracles
\[ f_{\mathcal{H}}(x) = \mathcal{H}(x) \]

Unstructured

Isogeny Shortcut
\[ f_{\Phi}(x) = \Phi(x) \]

Universal Functions
Obfuscation, FHE

Real-World

Theory-World
Enter Quantum Computing

- Structured
- Unstructured

Random Oracles

Isogeny Shortcut

\[ f_\Phi(x) = \Phi(x) \]

Universal Functions

Obfuscation, FHE

Real-World

Theory-World
Enter Quantum Computing

Structured

Unstructured

Random Oracles

\[ f_{\mathcal{H}}(x) = \mathcal{H}(x) \]

Isogeny Shortcut

\[ f_{\Phi}(x) = \Phi(x) \]

Universal Functions

Obfuscation, FHE
Can we construct efficient post-quantum timed cryptography?
Our Results
Our Results

• A new candidate “lattice-based” sequential function
Our Results

• A new candidate “lattice-based” sequential function

• Evidence of sequentiality
Our Results

• A new candidate “lattice-based” sequential function
  • Evidence of sequentiality
  • Cryptanalysis
Our Results

• A new candidate “lattice-based” sequential function
  • Evidence of sequentiality
  • Cryptanalysis
• Application: Simple proof of sequential work
Our Results

• A new candidate “lattice-based” sequential function
  • Evidence of sequentiality
  • Cryptanalysis
• Application: Simple proof of sequential work
  • Protocol & soundness analysis
Our Results

• A new candidate “lattice-based” sequential function
  • Evidence of sequentiality
  • Cryptanalysis
• Application: Simple proof of sequential work
  • Protocol & soundness analysis
• Open problems
A New Sequential Function
A New Sequential Function

- Input: a matrix $A \in \mathbb{Z}_q^{n \times m}$ and a vector $x \in \mathbb{Z}_q^n$

$m \approx n \cdot \log q$
A New Sequential Function

- Input: a matrix $A \in \mathbb{Z}^{n \times m}_q$ and a vector $x \in \mathbb{Z}^n_q$
- Compute:

$$m \approx n \cdot \log q$$
A New Sequential Function

- Input: a matrix \( A \in \mathbb{Z}_{q}^{n \times m} \) and a vector \( x \in \mathbb{Z}_{q}^{n} \)

- Compute:
  - The binary decomposition \( u = G^{-1}(x) \)

\[ m \approx n \cdot \log q \]
A New Sequential Function

- Input: a matrix $A \in \mathbb{Z}_{q}^{n \times m}$ and a vector $x \in \mathbb{Z}_{q}^{n}$

- Compute:
  - The binary decomposition $u = G^{-1}(x)$
  - Return $A \cdot u \mod q$
A New Sequential Function

- Input: a matrix $A \in \mathbb{Z}_{q}^{n \times m}$ and a vector $x \in \mathbb{Z}_{q}^{n}$

- Compute:
  - The binary decomposition $u = G^{-1}(x)$
  - Return $A \cdot u \mod q$

- Feed the output of the function as an input $T$ times

$m \approx n \cdot \log q$
A New Sequential Function

• Input: a matrix $A \in \mathbb{Z}_q^{n\times m}$ and a vector $x \in \mathbb{Z}_q^n$

• Compute:
  
  • The binary decomposition $u = G^{-1}(x)$
  
  • Return $A \cdot u \mod q$

• Feed the output of the function as an input T times

$$f_A(x) = y = A \cdot G^{-1}(x)$$
A New Sequential Function

- Input: a matrix $A \in \mathbb{Z}_{q}^{n \times m}$ and a vector $x \in \mathbb{Z}_{q}^{n}$

- Compute:
  - The binary decomposition $u = G^{-1}(x)$
  - Return $A \cdot u \mod q$

- Feed the output of the function as an input $T$ times

$$f_{A}(x) = y = A \cdot G^{-1}(x) \iff \begin{bmatrix} G \\ A \end{bmatrix} \cdot u = \begin{bmatrix} x \\ y \end{bmatrix} \quad u \in \{0,1\}^{m}$$

$m \approx n \cdot \log q$
The Sequentiality Assumption
The Sequentiality Assumption

• For uniform $A$ and $x$, the T-fold recursive application of $f_A$ is sequential
The Sequentiality Assumption

• For uniform $A$ and $x$, the T-fold recursive application of $f_A$ is sequential

• In other words, it takes parallel time $T$ to find $u$ such that
The Sequentiality Assumption

- For uniform $A$ and $x$, the $T$-fold recursive application of $f_A$ is sequential
- In other words, it takes parallel time $T$ to find $u$ such that

$$
\begin{bmatrix}
-G \\
A & -G \\
A & A \\
& & \ddots \\
& & & -G \\
& & & A
\end{bmatrix}
\cdot u =
\begin{bmatrix}
-x \\
0 \\
0 \\
\vdots \\
y
\end{bmatrix}
\quad u \in \{0,1\}^m
$$
The Strong Sequentiality Assumption

- For uniform $A$ and $x$, the T-fold recursive application of $f_A$ is sequential.

- In other words, it takes parallel time $T$ to find $u$ such that:

$$
\begin{bmatrix}
-G \\
A & -G \\
& A & -G \\
\end{bmatrix}
\cdot u =
\begin{bmatrix}
-x \\
0 \\
\vdots \\
0 \\
y
\end{bmatrix}
\Rightarrow u \approx \text{small}
$$
Evidence of Sequentiality
Evidence of Sequentiality

• The function $f_A$ is collision-resistant
Evidence of Sequentiality

• The function $f_A$ is collision-resistant
  • Simple reduction to the SIS problem w.r.t. $A$
Evidence of Sequentiality

• The function $f_A$ is collision-resistant
  • Simple reduction to the SIS problem w.r.t. $A$

• The function $f_A$ is uniformity-preserving (assuming LWE)
Evidence of Sequentiality

- The function $f_A$ is collision-resistant
  - Simple reduction to the SIS problem w.r.t. $A$
- The function $f_A$ is uniformity-preserving (assuming LWE)
  - $\{y : y \leftarrow \mathbb{Z}_q^n\} \approx \{A(x) : x \leftarrow \mathbb{Z}_q^n\}$
Evidence of Sequentiality

• The function $f_A$ is collision-resistant
  
  • Simple reduction to the SIS problem w.r.t. $A$

• The function $f_A$ is uniformity-preserving (assuming LWE)
  
  • $\{y : y \leftarrow \mathbb{Z}_q^n\} \approx \{A(x) : x \leftarrow \mathbb{Z}_q^n\}$
  
  • Consequently, so is its T-fold repetition
Evidence of Sequentiality

- The function $f_A$ is collision-resistant
  - Simple reduction to the SIS problem w.r.t. $A$

- The function $f_A$ is uniformity-preserving (assuming LWE)
  - $\{y : y \leftarrow \mathbb{Z}_q^n\} \approx \{A(x) : x \leftarrow \mathbb{Z}_q^n\}$
  - Consequently, so is its T-fold repetition

- More heuristic evidence & cryptanalysis (see paper)
Proofs of Sequential Work

\[ P \xrightarrow{A, x, y} V \xrightarrow{\{0,1\}} \]
Proofs of Sequential Work

- Succinctness: V’s work is $\sim \log T$
Proofs of Sequential Work

- Succinctness: V’s work is $\sim \log T$
- Soundness: No prover of depth $<< T$ can pass the verification
Self-Symmetry
Self-Symmetry

\[ A_T = \begin{bmatrix} -G \\ A \\ -G \\ A \\ \vdots \\ -G \\ A \end{bmatrix} \]
Self-Symmetry

\[ A_T = \begin{bmatrix} -G & A & -G \\ A & -G & A \\ \vdots & & \ddots \\ -G & A & -G \\ A & -G & A \\ \end{bmatrix} = \begin{bmatrix} A_t \\ -G \\ A \\ \end{bmatrix} \]

\[ T = 2t + 1 \]
The Protocol (Step 1)
The Protocol (Step 1)

- P sends the intermediate value $u_t$ to V
The Protocol (Step 1)

- P sends the intermediate value $u_t$ to V
- V checks that $u_t$ is small
The Protocol (Step 1)

- P sends the intermediate value $u_t$ to V
- V checks that $u_t$ is small
- The new relation is:
The Protocol (Step 1)

- P sends the intermediate value $\mathbf{u}_t$ to V
- V checks that $\mathbf{u}_t$ is small
- The new relation is:

$$
\begin{align*}
A_t \cdot 
\begin{bmatrix}
\mathbf{u}_0 & \mathbf{u}_{t+1} \\
\vdots & \vdots \\
\mathbf{u}_{t-1} & \mathbf{u}_{T-1}
\end{bmatrix}
= 
\begin{bmatrix}
-x_0 & -A \cdot \mathbf{u}_t \\
0 & 0 \\
\vdots & \vdots \\
0 & 0 \\
-G \cdot \mathbf{u}_t & x_T
\end{bmatrix}
\end{align*}
$$
The Protocol (Step 2)
The Protocol (Step 2)

- V sends a small $r$ to P
The Protocol (Step 2)

• V sends a small $r$ to P

• The new relation is:
The Protocol (Step 2)

• V sends a small $r$ to P

• The new relation is:

$$A_t \cdot \begin{bmatrix} u_0 + u_{t+1} \cdot r \\ \vdots \\ u_{t-1} + u_{T-1} \cdot r \end{bmatrix} = \begin{bmatrix} -x_0 - A \cdot u_t \cdot r \\ 0 \\ \vdots \\ 0 \\ -G \cdot u_t + x_T \cdot r \end{bmatrix}$$
The Protocol (Step 2)

- V sends a small $r$ to P
- The new relation is:

$$A_t \cdot \begin{bmatrix} u_0 + u_{t+1} \cdot r \\ \vdots \\ u_{t-1} + u_{T-1} \cdot r \end{bmatrix} = \begin{bmatrix} -x_0 - A \cdot u_t \cdot r \\ 0 \\ \vdots \\ 0 \\ -G \cdot u_t + x_T \cdot r \end{bmatrix}$$

- Dimension halved, recurse!
The Protocol (Step 2)

- V sends a small $r$ to P
- The new relation is:

\[
A_t \cdot \begin{bmatrix}
    u_0 + u_{t+1} \cdot r \\
    \vdots \\
    u_{t-1} + u_{T-1} \cdot r
\end{bmatrix} = \begin{bmatrix}
    -x_0 - A \cdot u_t \cdot r \\
    0 \\
    \vdots \\
    0 \\
    -G \cdot u_t + x_T \cdot r
\end{bmatrix}
\]

- Dimension halved, recurse!
- Verifier’s runtime $\sim \log(T)$
Soundness
Soundness

- (2, …, 2)-special soundness: Given a binary tree of accepting transcripts, one can recover a valid witness
Soundness

• (2, …, 2)-special soundness: Given a binary tree of accepting transcripts, one can recover a valid witness

• Can be shown if we sample the challenges from the appropriate domain (subtractive sets)
Soundness

- (2, …, 2)-special soundness: Given a binary tree of accepting transcripts, one can recover a valid witness

- Can be shown if we sample the challenges from the appropriate domain (subtractive sets)

- [AF’22] There exists a depth-preserving extractor for the parallel-repeated protocol
Soundness

• (2, …, 2)-special soundness: Given a binary tree of accepting transcripts, one can recover a valid witness

• Can be shown if we sample the challenges from the appropriate domain (subtractive sets)

• [AF’22] There exists a depth-preserving extractor for the parallel-repeated protocol

• Can extract a valid transcript in time $o(T)$
Soundness

• (2, …, 2)-special soundness: Given a binary tree of accepting transcripts, one can recover a valid witness

• Can be shown if we sample the challenges from the appropriate domain (subtractive sets)

• [AF’22] There exists a depth-preserving extractor for the parallel-repeated protocol

• Can extract a valid transcript in time o(T)

• Contradiction!
Open Problems
Open Problems

• Efficient “lattice-based” VDF?
Open Problems

• Efficient “lattice-based” VDF?
  • Candidate construction: Valerio’s talk!
Open Problems

• Efficient “lattice-based” VDF?
  • Candidate construction: Valerio’s talk!

• More evidence of sequentiality?
Open Problems

- Efficient “lattice-based” VDF?
  - Candidate construction: Valerio’s talk!
- More evidence of sequentiality?
- More applications?
Open Problems

- Efficient “lattice-based” VDF?
  - Candidate construction: Valerio’s talk!

- More evidence of sequentiality?

- More applications?

- Post-quantum security?
Open Problems

• Efficient "lattice-based" VDF?
  • Candidate construction: Valerio’s talk!

• More evidence of sequentiality?

• More applications?

• Post-quantum security?

Thank you!