A Subexponential Quantum Algorithm for the Semidirect Discrete Logarithm Problem

Christopher Battarbee
Joint with D. Kahrobaei, L. Perret, S. F. Shahandashti

University of York, CUNY, Sorbonne

4^{th} PQC Standardization Conference 2022

30 November 2022

Motivation

Group-based Cryptography

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography
- Diverse roster of computational problems

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography
- Diverse roster of computational problems
- Quantum hardness of these problems not well understood.

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography
- Diverse roster of computational problems
- Quantum hardness of these problems not well understood.

Non-interactive Key Exchange (NIKE)

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography
- Diverse roster of computational problems
- Quantum hardness of these problems not well understood.

Non-interactive Key Exchange (NIKE)

- Use cases for resource-constrained settings vs KEMs

Motivation

Group-based Cryptography

- Relatively understudied family in post-quantum cryptography
- Diverse roster of computational problems
- Quantum hardness of these problems not well understood.

Non-interactive Key Exchange (NIKE)

- Use cases for resource-constrained settings vs KEMs
- Not many such schemes available

A Novel Exponentiation

Definition

- Let G a finite, non-abelian (semi)group and $End(G)$ its induced semigroup of endomorphisms
- Each $(g, \phi) \in G \times End(G)$ induces a function $s_{g,\phi} : \mathbb{N} \to G$

$$s_{g,\phi}(x) = \phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$$

- $s_{g,\phi}$ is efficiently calculable via the semidirect product.

A Novel Exponentiation

Definition

- Let G a finite, non-abelian (semi)group and $End(G)$ its induced semigroup of endomorphisms
- Each $(g, \phi) \in G \times End(G)$ induces a function $s_{g,\phi} : \mathbb{N} \to G$

$$s_{g,\phi}(x) = \phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$$

- $s_{g,\phi}$ is efficiently calculable via the semidirect product.

$$\phi^{x-1}(g) \cdot \ldots \cdot g$$

A Novel Exponentiation

Definition

- Let G a finite, non-abelian (semi)group and $End(G)$ its induced semigroup of endomorphisms
- Each $(g, \phi) \in G \times End(G)$ induces a function $s_{g,\phi} : \mathbb{N} \to G$

$$s_{g,\phi}(x) = \phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$$

- $s_{g,\phi}$ is efficiently calculable via the semidirect product.

$$\phi^{x-1}(g) \cdot \ldots \cdot g \longrightarrow \boxed{\phi^{y}} \longrightarrow \phi^{x+y-1}(g) \cdot \ldots \cdot \phi^{y}(g)$$

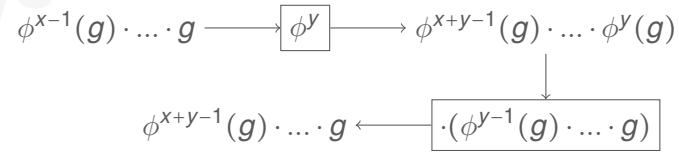
A Novel Exponentiation

Definition

- Let G a finite, non-abelian (semi)group and $End(G)$ its induced semigroup of endomorphisms
- Each $(g, \phi) \in G \times End(G)$ induces a function $s_{g,\phi} : \mathbb{N} \to G$

$$s_{g,\phi}(x) = \phi^{x-1}(g) \cdot \ldots \cdot \phi(g) \cdot g$$

- $s_{g,\phi}$ is efficiently calculable via the semidirect product.



A Novel Exponentiation

Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

A Novel Exponentiation

Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

There is a function $* : \mathbb{N} \times \mathcal{X}_{g,\phi} \to \mathcal{X}_{g,\phi}$, pronounced *step*, such that

$$j * s_{g,\phi}(i) = s_{g,\phi}(i + j)$$

A Novel Exponentiation

Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

There is a function $* : \mathbb{N} \times \mathcal{X}_{g,\phi} \to \mathcal{X}_{g,\phi}$, pronounced *step*, such that

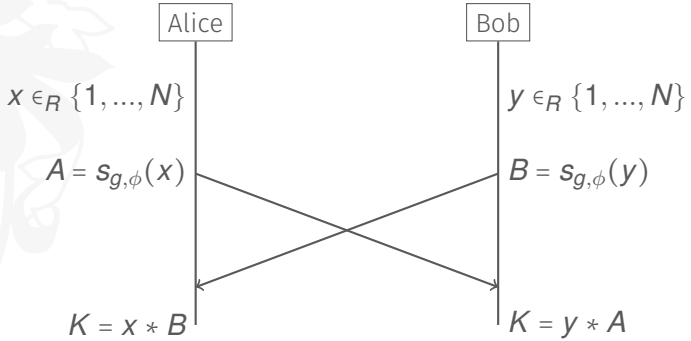
$$j * s_{g,\phi}(i) = s_{g,\phi}(i + j)$$

Indeed

$$y * s_{g,\phi}(x) = s_{g,\phi}(x + y) = x * s_{g,\phi}(y)$$

Semidirect Product Key Exchange (SDPKE)

Public parameters:
$G, (g, \phi), N = |\mathcal{X}_{g,\phi}|$



Habeeb, Kahrobaei, Koupparis, and Shpilrain. "Public key exchange using semidirect product of (semi) groups". In: *ACNS*. Springer

Non-interactive SDPKE)

Public parameters:

$G, (g, \phi), N = |\mathcal{X}_{g,\phi}|$

Alice	Bob
$sk_A = x \in_R \{1, ..., N\}$	$sk_B = y \in_R \{1, ..., N\}$
$pk_A = s_{g,\phi}(x)$	$pk_B = s_{g,\phi}(y)$
$K = x * pk_B$	$K = y * pk_A$

SDLP, SCDH

Suppose $N = \#\{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

SDLP, SCDH

Suppose $N = \#\{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

Definition (SDLP)

Let $(g, \phi) \in G \times End(G)$ and x sampled uniformly at random from $\{1, ...N\}$. One solves the *Semidirect Discrete Logarithm Problem* (SDLP) if one can recover x from $(g, \phi), s_{g,\phi}(x)$.

SDLP, SCDH

Suppose $N = \#\{s_{g,\phi}(i) : i \in \mathbb{N}\}$.

Definition (SDLP)

Let $(g, \phi) \in G \times End(G)$ and x sampled uniformly at random from $\{1, ...N\}$. One solves the *Semidirect Discrete Logarithm Problem* (SDLP) if one can recover x from $(g, \phi), s_{g,\phi}(x)$.

Definition (SCDH)

Let $(g, \phi) \in G \times End(G)$ and x, y sampled uniformly at random from $\{1, ..., N\}$. One solves *Semidirect Computational Diffie-Hellman* (SCDH) if one can recover $s_{g,\phi}(x + y)$ from $(g, \phi), s_{g,\phi}(x), s_{g,\phi}(y)$.

The Vectorisation Problem

Definition (Group Actions)

A *group action* is a triple $(G, X, \star)$ where G is a group, X is a set, and $\star : G \times X \to X$ is a function such that:

- $1 \star x = x$ for all $x \in X$
- $(a + b) \star x = a \star (b \star x)$ for all $a, b \in G$, $x \in X$

The Vectorisation Problem

Definition (Group Actions)

A *group action* is a triple $(G, X, \star)$ where G is a group, X is a set, and $\star : G \times X \to X$ is a function such that:

- $1 \star x = x$ for all $x \in X$
- $(a + b) \star x = a \star (b \star x)$ for all $a, b \in G$, $x \in X$

Definition (Vectorisation Problem)

Let $(G, X, \star)$ a finite group action. Given $x, y \in X$, the *Vectorisation Problem* is to find $g \in G$ such that $g \star x = y$.

Jean-Marc Couveignes. "Hard homogeneous spaces". In: *Cryptology ePrint Archive* (2006)

Contributions

- We have a function, $s_{g,\phi}$, that behaves like the exponentiation map

Contributions

- We have a function, $s_{g,\phi}$, that behaves like the exponentiation map
- This function induces a Diffie-Hellman-like NIKE, and discrete log-like computational problems SDLP, SCDH

Contributions

- We have a function, $s_{g,\phi}$, that behaves like the exponentiation map
- This function induces a Diffie-Hellman-like NIKE, and discrete log-like computational problems SDLP, SCDH
- There is a discrete-log like computational problem in the context of group actions called the Vectorisation Problem.

Contributions

- We have a function, $s_{g,\phi}$, that behaves like the exponentiation map
- This function induces a Diffie-Hellman-like NIKE, and discrete log-like computational problems SDLP, SCDH
- There is a discrete-log like computational problem in the context of group actions called the Vectorisation Problem.

- We construct a quantum algorithm that solves SDLP in subexponential time

Contributions

- We have a function, $s_{g,\phi}$, that behaves like the exponentiation map
- This function induces a Diffie-Hellman-like NIKE, and discrete log-like computational problems SDLP, SCDH
- There is a discrete-log like computational problem in the context of group actions called the Vectorisation Problem.

- We construct a quantum algorithm that solves SDLP in subexponential time
- We show the following three computational problems are quantum equivalent: SDLP, SCDH, and Couveignes' *Vectorisation Problem*

Tail and Cycle

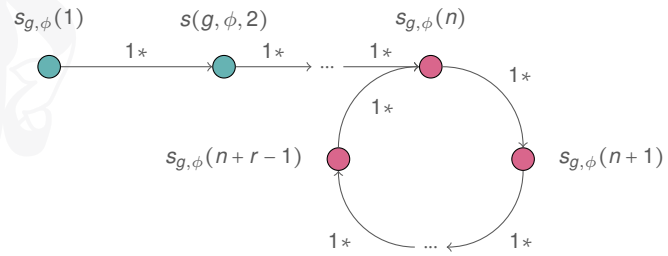
Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$. There is a function $* : \mathbb{N} \times \mathcal{X}_{g,\phi}$ such that

$$j * s_{g,\phi}(i) = s_{g,\phi}(i + j)$$

Tail and Cycle

Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$. There is a function $* : \mathbb{N} \times \mathcal{X}_{g,\phi}$ such that

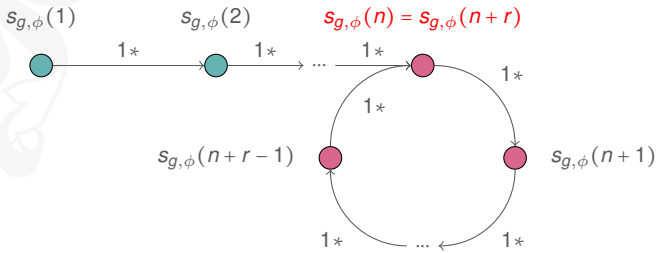
$$j * s_{g,\phi}(i) = s_{g,\phi}(i + j)$$



Tail and Cycle

Let $\mathcal{X}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{N}\}$. There is a function $* : \mathbb{N} \times \mathcal{X}_{g,\phi} \to \mathcal{X}_{g,\phi}$ such that

$$j * s_{g,\phi}(i) = s_{g,\phi}(i + j)$$



$$s_{g,\phi}(n + r + 1) = 1 * s_{g,\phi}(n + r) = 1 * s_{g,\phi}(n) = s_{g,\phi}(n + 1)$$

Structure of the Exponents

Each pair (g, ϕ) induces:

Structure of the Exponents

Each pair (g, ϕ) induces:

- $n_{g,\phi}$ - the *index* - start of loop

Structure of the Exponents

Each pair (g, ϕ) induces:

- $n_{g,\phi}$ - the *index* - start of loop
- $r_{g,\phi}$ - the *period* - length of loop

Structure of the Exponents

Each pair (g, ϕ) induces:

- $n_{g,\phi}$ - the *index* - start of loop
- $r_{g,\phi}$ - the *period* - length of loop
- A set $\mathcal{T}_{g,\phi} = \{g, ..., s_{g,\phi}(n-1)\}$ - the *tail* - the values with no periodic behaviour

Structure of the Exponents

Each pair (g, ϕ) induces:

- $n_{g,\phi}$ - the *index* - start of loop
- $r_{g,\phi}$ - the *period* - length of loop
- A set $\mathcal{T}_{g,\phi} = \{g, ..., s_{g,\phi}(n-1)\}$ - the *tail* - the values with no periodic behaviour
- A set $\mathcal{C}_{g,\phi} = \{s_{g,\phi}(n), ..., s_{g,\phi}(n+r-1)\}$ - the *cycle* - a set invariant under $r*$.

Structure of the Exponents

Each pair (g, ϕ) induces:

- $n_{g,\phi}$ - the *index* - start of loop
- $r_{g,\phi}$ - the *period* - length of loop
- A set $\mathcal{T}_{g,\phi} = \{g, ..., s_{g,\phi}(n-1)\}$ - the *tail* - the values with no periodic behaviour
- A set $\mathcal{C}_{g,\phi} = \{s_{g,\phi}(n), ..., s_{g,\phi}(n+r-1)\}$ - the *cycle* - a set invariant under $r*$.

Theorem

A cyclic group of size $r_{g,\phi}$, denoted $\mathbb{N}_r$, acts on the cycle $\mathcal{C}_{g,\phi}$.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

- Reasonably well known[1] that the Vectorisation Problem admits quantum subexponential algorithms.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

- Reasonably well known[1] that the Vectorisation Problem admits quantum subexponential algorithms.
- Given (g, ϕ) and $s_{g,\phi}(x)$, if we can find an efficient quantum reduction to the Vectorisation Problem we are therefore done.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

- Reasonably well known[1] that the Vectorisation Problem admits quantum subexponential algorithms.
- Given (g, ϕ) and $s_{g,\phi}(x)$, if we can find an efficient quantum reduction to the Vectorisation Problem we are therefore done.
- Some technicality since $s_{g,\phi}(x)$ might be in the tail which does not admit a known group action.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

- Reasonably well known[1] that the Vectorisation Problem admits quantum subexponential algorithms.
- Given (g, ϕ) and $s_{g,\phi}(x)$, if we can find an efficient quantum reduction to the Vectorisation Problem we are therefore done.
- Some technicality since $s_{g,\phi}(x)$ might be in the tail which does not admit a known group action.
- In practice require knowledge of $n_{g,\phi}, r_{g,\phi}$ - assume we know them for now.

[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP

Theorem

One can solve SDLP in quantum subexponential time.

- Reasonably well known[1] that the Vectorisation Problem admits quantum subexponential algorithms.
- Given (g, ϕ) and $s_{g,\phi}(x)$, if we can find an efficient quantum reduction to the Vectorisation Problem we are therefore done.
- Some technicality since $s_{g,\phi}(x)$ might be in the tail which does not admit a known group action.
- In practice require knowledge of $n_{g,\phi}, r_{g,\phi}$ - assume we know them for now.
- $s_{g,\phi}(x)$ is in the cycle if and only if $r_{g,\phi} \star s_{g,\phi}(x) = s_{g,\phi}(x)$

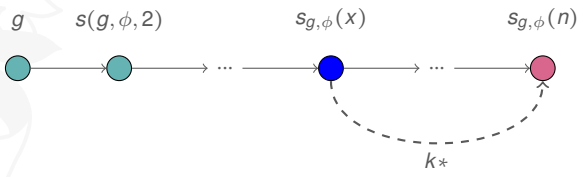
[1] Andrew Childs, David Jao, and Vladimir Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

Solving SDLP - Case 1

If $s_{g,\phi}(x)$ is in the tail:

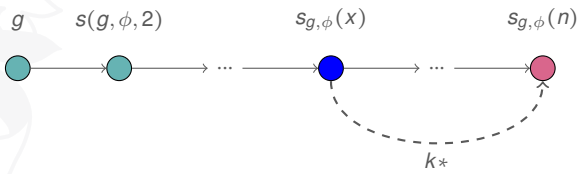
Solving SDLP - Case 1

If $s_{g,\phi}(x)$ is in the tail:



Solving SDLP - Case 1

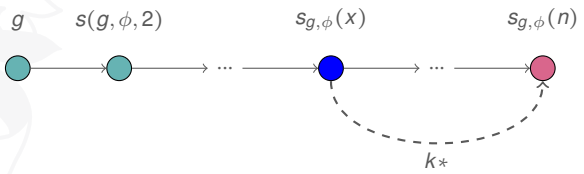
If $s_{g,\phi}(x)$ is in the tail:



- Find the smallest j such that $j * s_{g,\phi}(x)$ is invariant under $r*$ - can be done efficiently with binary search

Solving SDLP - Case 1

If $s_{g,\phi}(x)$ is in the tail:



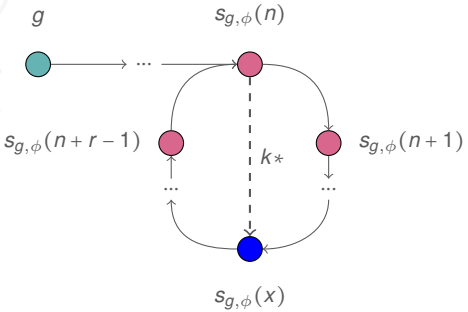
- Find the smallest j such that $j * s_{g,\phi}(x)$ is invariant under $r*$ - can be done efficiently with binary search
- This value, say k, is such that $n - k = x$.

Solving SDLP - Case 2

If $s_{g,\phi}(x)$ is in the cycle:

Solving SDLP - Case 2

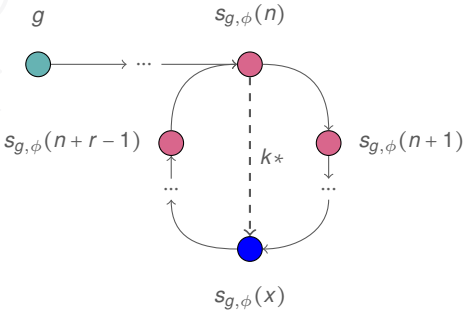
If $s_{g,\phi}(x)$ is in the cycle:



- Since a group acts on the cycle we can use a Vectorisation Problem oracle on $s_{g,\phi}(n), s_{g,\phi}(x)$

Solving SDLP - Case 2

If $s_{g,\phi}(x)$ is in the cycle:



- Since a group acts on the cycle we can use a Vectorisation Problem oracle on $s_{g,\phi}(n), s_{g,\phi}(x)$
- This group element tells us k such that $k * s_{g,\phi}(n) = s_{g,\phi}(x)$, so recover $x = n + k$.

The Quantum Part

How do we know $r_{g,\phi}$, $n_{g,\phi}$?

The Quantum Part

How do we know $r_{g,\phi}$, $n_{g,\phi}$?

$s_{g,\phi}(i)$ is $r_{g,\phi}$-periodic on $[n_{g,\phi}, \infty)$ - quantum algorithms are good at detecting periodicity!

The Quantum Part

How do we know $r_{g,\phi}, n_{g,\phi}$?

$s_{g,\phi}(i)$ is $r_{g,\phi}$-periodic on $[n_{g,\phi}, \infty)$ - quantum algorithms are good at detecting periodicity!

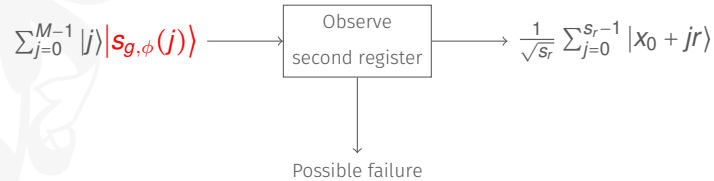
$$\sum_{j=0}^{M-1} |j\rangle |s_{g,\phi}(j)\rangle$$

Andrew M Childs and Gábor Ivanyos. “Quantum computation of discrete logarithms in semigroups”. In: *Journal of Mathematical Cryptology* 8.4 (2014), pp. 405–416

The Quantum Part

How do we know $r_{g,\phi}, n_{g,\phi}$?

$s_{g,\phi}(i)$ is $r_{g,\phi}$-periodic on $[n_{g,\phi}, \infty)$ - quantum algorithms are good at detecting periodicity!

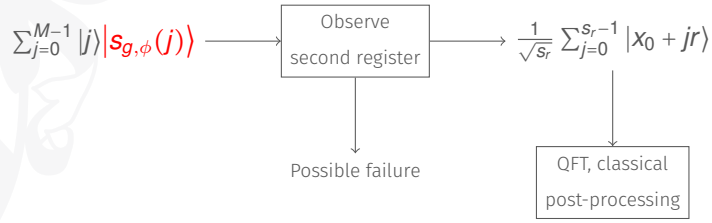


Andrew M Childs and Gábor Ivanyos. “Quantum computation of discrete logarithms in semigroups”. In: *Journal of Mathematical Cryptology* 8.4 (2014), pp. 405–416

The Quantum Part

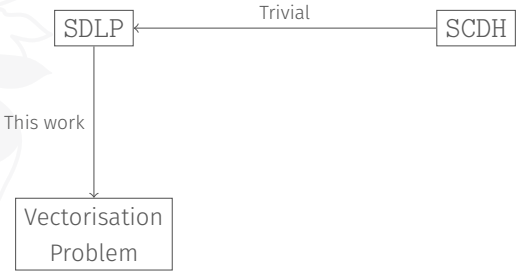
How do we know $r_{g,\phi}, n_{g,\phi}$?

$s_{g,\phi}(i)$ is $r_{g,\phi}$-periodic on $[n_{g,\phi}, \infty)$ - quantum algorithms are good at detecting periodicity!

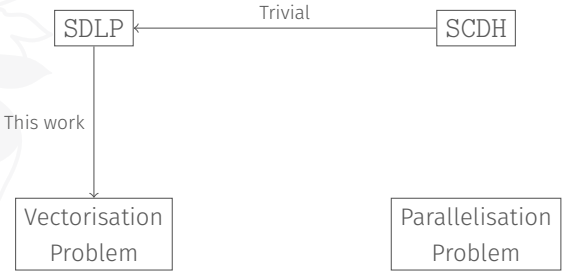


Andrew M Childs and Gábor Ivanyos. "Quantum computation of discrete logarithms in semigroups". In: *Journal of Mathematical Cryptology* 8.4 (2014), pp. 405–416

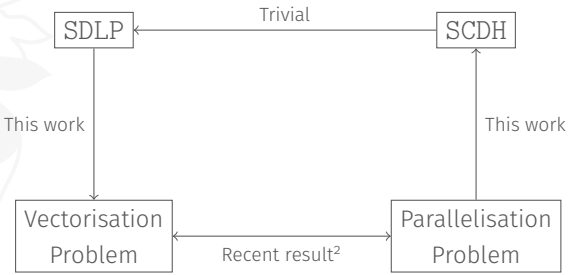
The Landscape of Problems



The Landscape of Problems



The Landscape of Problems



[2] Hart Montgomery and Mark Zhandry. "Full quantum equivalence of group action DLog and CDH, and more". In: *ASIACRYPT* (2022).

So What?

- We provide a much sharper classification of the quantum difficulty of a group-based computational problem

So What?

- We provide a much sharper classification of the quantum difficulty of a group-based computational problem
- We increase the diversity of plausibly post-quantum NIKEs; other examples include CSIDH

So What?

- We provide a much sharper classification of the quantum difficulty of a group-based computational problem
- We increase the diversity of plausibly post-quantum NIKEs; other examples include CSIDH
- Our framework is a rare non-trivial example of a cryptographically useful group action

So What?

- We provide a much sharper classification of the quantum difficulty of a group-based computational problem
- We increase the diversity of plausibly post-quantum NIKEs; other examples include CSIDH
- Our framework is a rare non-trivial example of a cryptographically useful group action
- Lots of further study required!

The End.

Thank you.