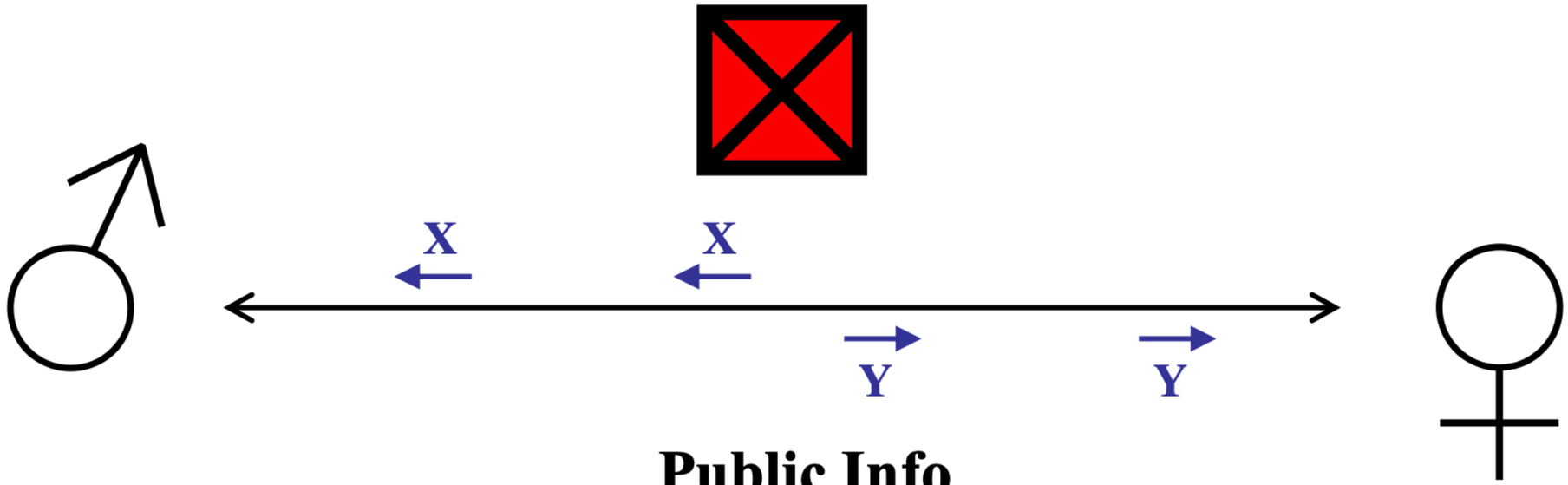
Non-commutative Diffie-Hellman



Public Info

G: non-abelian f.g. group with solvable WP

$g \in G$; $S, T < G$ s.t. $[S, T] = \{1\}$

$y \in T$

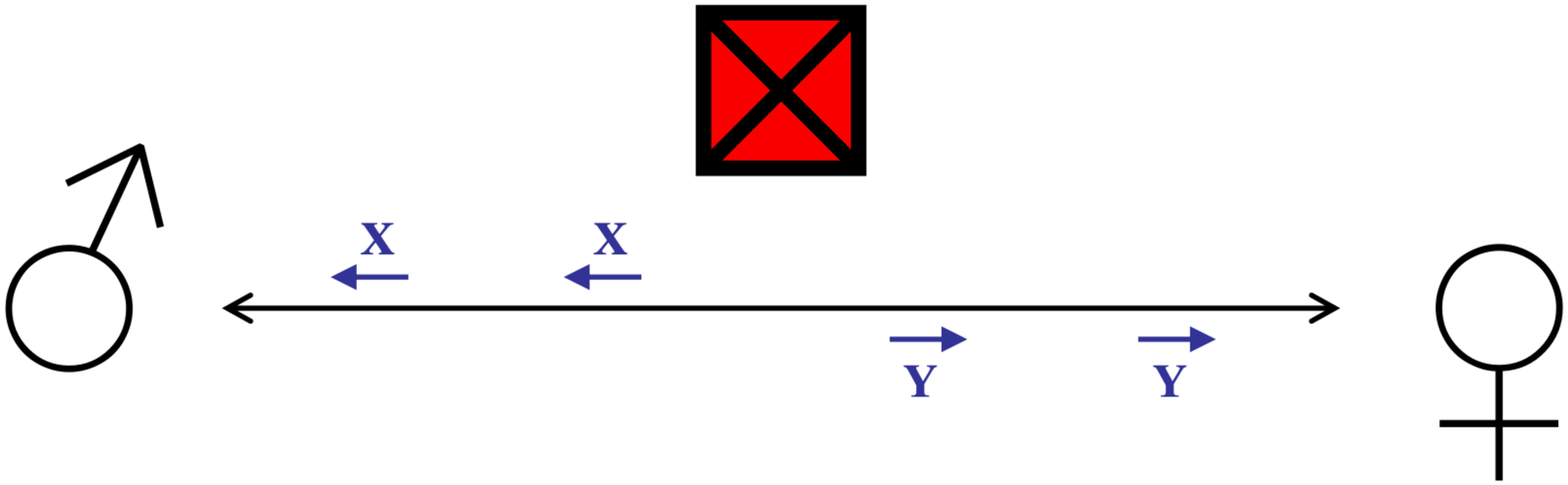
$Y = g^y = y^{-1}\, g\, y$

$x \in S$

$X = g^x = x^{-1}\, g\, x$

G: non-abelian f.g. group with solvable WP

$g \in G$; $S, T < G$ s.t. $[S, T] = \{1\}$



$k' = X^y = (g^x)^y = g^{x\,y} = (g^y)^x = Y^x = k$

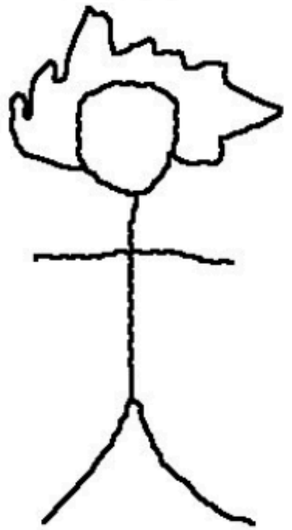
$[x, y]=1 \Rightarrow g^{x\,y} = g^{y\,x}$

To Break: Need to solve CP

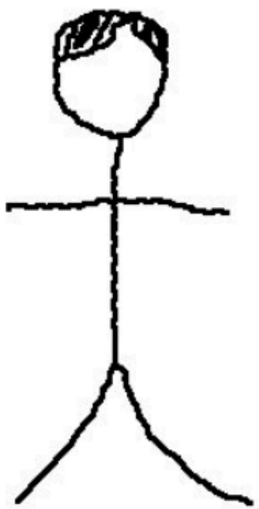
(Find x , y from g, $X = g^x$, $Y = g^y$)

Introduction to AAG

Alice



Bob



public:

Alice's public set: $\overline{a} = (a_1, \ldots, a_{N_1})$

Bob's public set: $\overline{b} = (b_1, \ldots, b_{N_2})$

$b_i' = A^{-1} b_i A \quad b_i \in \overline{b} \quad i = 1, \ldots, N_2$ →

← $a_i' = B^{-1} a_i B \quad a_i \in \overline{a} \quad i = 1, \ldots, N_1$

Alice's private key:

$A = a_{s_1}^{\varepsilon_1} \ldots a_{s_L}^{\varepsilon_L}$

$a_{s_i} \in \overline{a}, \varepsilon_i = \pm 1$

Bob's private key:

$B = b_{t_1}^{\delta_1} \ldots b_{t_L}^{\delta_L}$

$b_{t_i} \in \overline{b}, \delta_i = \pm 1$

AAG Common Key

Alice computes the common key by:

$$\begin{aligned} K_A &= A^{-1} a'^{\varepsilon_1}_{s_1} \dots a'^{\varepsilon_L}_{s_L} \\ &= A^{-1} (B^{-1} a_{s_1} B)^{\varepsilon_1} \dots (B^{-1} a_{s_L} B)^{\varepsilon_L} \\ &= A^{-1} B^{-1} a^{\varepsilon_1}_{s_1} \dots a^{\varepsilon_L}_{s_L} B \\ &= A^{-1} B^{-1} A B = K \end{aligned}$$

Similarly, Bob computes $K_B = B^{-1} b'^{\delta_1}_{t_1} \dots b'^{\delta_L}_{t_L} = B^{-1} A^{-1} B A$, then the common key is $K = K_B^{-1}$

Length-Based Attack

Eve sees public information: $\overline{a} = (a_1, \ldots, a_{N_1})$, $\overline{b} = (b_1, \ldots, b_{N_2})$ and $\overline{b'} = (b'_1, \ldots, b'_{N_2})$ such that $b'_i = b_i^A$ for $i = 1, \ldots, N_2$

Want to guess $A' \in \langle a_1, \ldots, a_{N_1} \rangle$ such that $b'_i = b_i^{A'}$ (or symmetrically, B').

Length-Based Attack (cont.)

$$
\begin{array}{c}
b_i \\
\downarrow \\
a_{s_1}^{-\varepsilon_1} b_i a_{s_1}^{\varepsilon_1} \\
\downarrow \\
a_{s_2}^{-\varepsilon_2} a_{s_1}^{-\varepsilon_1} b_i a_{s_1}^{\varepsilon_1} a_{s_2}^{\varepsilon_2} \\
\downarrow \\
\vdots \\
\downarrow \\
a_{s_L}^{-\varepsilon_L} \ldots a_{s_2}^{-\varepsilon_2} a_{s_1}^{-\varepsilon_1} b_i a_{s_1}^{\varepsilon_1} a_{s_2}^{\varepsilon_2} \ldots a_{s_L}^{\varepsilon_L}
\end{array}
$$

Idea: going back from bottom to top, at each step, length is decreased.

If conjugated tuple is the same as $\overline{b}$ then stop and output conjugator by reversing the sequence of conjugating factors.

Computational Complexity and Group Theory

Glasgow Math. J. **61** (2019) 251–269. © Glasgow Mathematical Journal Trust 2018.
doi:10.1017/S0017089518000198.

ON THE CONJUGACY PROBLEM IN CERTAIN METABELIAN GROUPS

JONATHAN GRYAK

Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA e-mail: gryakj@med.umich.edu

DELARAM KAHROBAEI

CUNY Graduate Center, PhD Program in Computer Science and NYCCT, Mathematics Department, City University of New York, New York, NY, USA e-mail: dkahrobaei@gc.cuny.edu

and CONCHITA MARTINEZ-PEREZ

Department of Mathematics, University of Zaragoza, Zaragoza, Spain e-mail: conmar@unizar.es

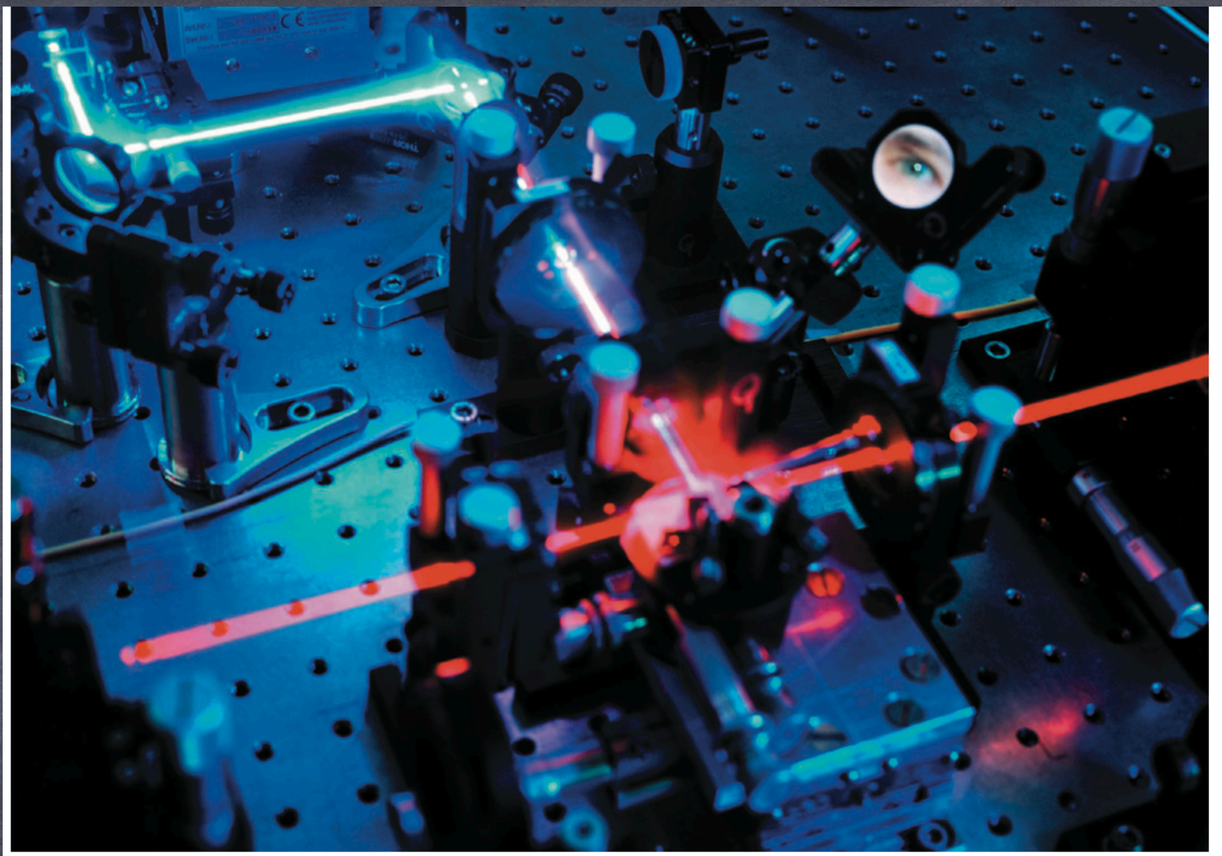
(Received 19 June 2017; revised 14 December 2017; accepted 13 April 2018; first published online 20 June 2018)

Abstract. We analyze the computational complexity of an algorithm to solve the conjugacy search problem in a certain family of metabelian groups. We prove that in general the time complexity of the conjugacy search problem for these groups is at most exponential. For a subfamily of groups, we prove that the conjugacy search problem is polynomial. We also show that for a different subfamily the conjugacy search problem reduces to the discrete logarithm problem.

Open Problems

- Finding the platform which is post-quantum and secure against attacks like Length-based attack, etc

Post-quantum Hash Functions and Blockchain Technology



Quantum cryptography equipment, which uses the principle of entanglement to encode data that only the sender and receiver can access.

Quantum computers put blockchain security at risk

Bitcoin and other cryptocurrencies will founder unless they integrate quantum technologies, warn **Aleksey K. Fedorov**, **Evgeniy O. Kiktenko** and **Alexander I. Lvovsky**.

By 2025, up to 10% of global gross domestic product is likely to be stored on blockchains[1]. A blockchain is a digital tool that uses cryptography techniques to protect information from unauthorized changes. It lies at the root of the Bitcoin cryptocurrency[2]. Blockchain-related products are used everywhere from finance and manufacturing to health care, in a market worth more than US$150 billion.

When information is money, data security, transparency and accountability are crucial. A blockchain is a secure digital record, or ledger. It is maintained collectively by users around the globe, rather than by one central administration. Decisions such as whether to add an entry (or block) to the ledger are based on consensus — so personal trust ▶

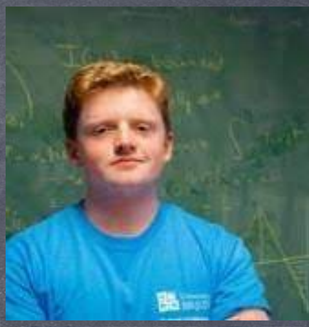
New Frontiers in Research Fund
Fonds Nouvelles frontières en recherche

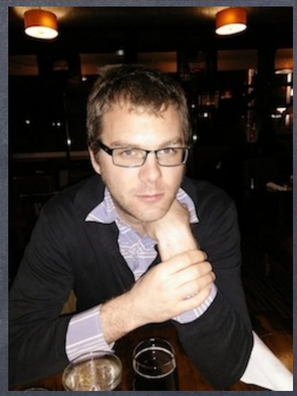
Office of Naval Research
Science & Technology

Advances in Mathematics of Communications

Post-quantum hash functions using $\mathrm{SL}_n(\mathbb{F}_p)$

Corentin Le Coz, Technion – Israel Institute of Technology
Christopher Battarbee, University of York
Ramón Flores, University of Seville
Thomas Koberda, University of Virginia
Delaram Kahrobaei, Queens College, CUNY, University of York



Abstract

We define new families of Tillich-Zémor hash functions, using higher dimensional special linear groups over finite fields as platforms.
The Cayley graphs of these groups combine fast mixing properties and high girth, which together give rise to good preimage and collision resistance of the corresponding hash functions. We justify the claim that the resulting hash functions are post-quantum secure.

What is a hash function?

Definition

A **hash function** is a map $\varphi : \mathbb{N} \to F$, with $\#F < \infty$, such that:

- *(preimage resistance)* given a random $x \in F$, it is hard to find $m \in \mathbb{N}$ such that $\varphi(m) = x$,
- *(collision resistance)* it is hard to find $m, m' \in \mathbb{N}$ such that $\varphi(m) = \varphi(m')$.

The set F is called the **platform** of the hash function φ.

Motivations: blockchains, post-quantum cryptography, NIST, etc.

Group-based Tillich-Zémor hash functions

[Tillich-Zémor '94], [Charles-Lauter-Goren '08]
Let G be a finite group. Let $S \subset G$, and $k := \#S$.

A first attempt: Let $\sigma : \{0, 1, \ldots, k-1\} \to S$ be a bijection.

$$\begin{array}{rccc} \varphi : & \mathbb{N} & \longrightarrow & G \\ & m = i_1 i_2 \ldots i_l \text{ (in base } k) & \longmapsto & \varphi(m) = \sigma(i_1) \cdot \sigma(i_2) \cdots \sigma(i_l) \end{array}$$

Problem:

Fact

If we assume that S is symmetric ($\forall x \in S$, $x^{-1} \in S$), we can find collisions easily.

Proof.

Say we have $\sigma(i) = \sigma(j)^{-1}$. Then, both $m = (ij)^{10}$ and $m' = i^{10} j^{10}$ are mapped to the identity element of G. □

Group-based Tillich-Zémor hash functions

[Tillich-Zémor '94], [Charles-Lauter-Goren '08]
Let G be a finite group. Let $S \subset G$, and $k := \#S$.
We will always assume:

- S is symmetric: $S = S^{-1}$,
- S generates G: $\langle S \rangle = G$.

Let $\sigma\colon S \times \{0, 1, \ldots, k-2\} \to S$ be such that for every $s \in S$, the map $\sigma(s, \cdot)$ is a bijection between $\{0, 1, \ldots, k-2\}$ and $S \setminus s^{-1}$.

Definition

The hash function φ associated with (G, S, σ) is defined as follows. Given $m = i_1 i_2 \ldots i_l$ (in base $k-1$), we define inductively:

- $s_1 = \sigma(s, i_1)$, for a fixed $s \in S$,
- $s_\lambda = \sigma(s_{\lambda-1}, i_\lambda)$, for every $\lambda \in \{2, \ldots, l\}$.

We then define: $\varphi(m) := s_1 \cdot s_2 \cdots s_l$.

A geometric interpretation

Definition

Given $G = \langle S \rangle$, we define the (directed) Cayley graph $\mathrm{Cay}(G, S) = (V, E)$ as follows:

- $V = G$,
- E contains the edge: $g \xrightarrow{s} gs$, for every $(g, s) \in G \times S$.

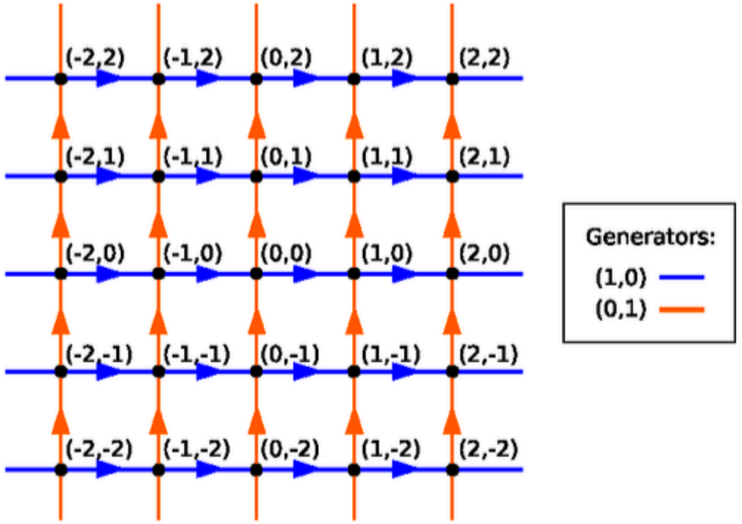


Figure: Cayley graph of $\mathbb{Z}^2$

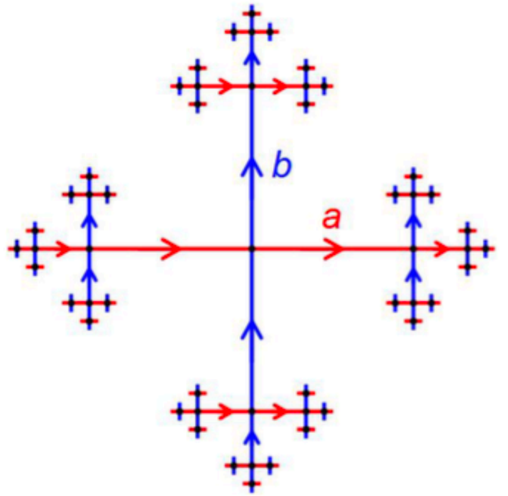


Figure: Cayley graph of F_2

Collision resistance and loops

Proposition

The problem of finding a loop in $\mathrm{Cay}(G, S)$ *reduces to the collision problem for* φ.

Proof.

Given $m \neq m'$ such that $\varphi(m) = \varphi(m')$, we have two distinct paths in $\mathrm{Cay}(G, S)$ joining id and $\varphi(m)$:

$$\mathrm{id} \xrightarrow{s_1} s_1 \xrightarrow{s_2} \ldots \xrightarrow{s_\lambda} s_1 \cdots s_\lambda \xrightarrow{s_{\lambda+1}} \ldots \xrightarrow{s_l} s_1 \cdots s_l = \varphi(m)$$

$$\mathrm{id} \xleftarrow{s_1^{-1}} s_1 \xleftarrow{s_2^{-1}} \ldots \xleftarrow{s_\lambda^{-1}} s_1 \cdots s_\lambda \xleftarrow{s_{\lambda+1}^{-1}} \ldots \xleftarrow{s_l^{-1}} s_1 \cdots s_l = \varphi(m)$$

$$\mathrm{id} \xrightarrow{s'_1} s'_1 \xrightarrow{s'_2} \ldots \xrightarrow{s'_\lambda} s'_1 \cdots s'_\lambda \xrightarrow{s'_{\lambda+1}} \ldots \xrightarrow{s'_l} s'_1 \cdots s'_l = \varphi(m)$$

We obtain a non-trivial loop in $\mathrm{Cay}(G, S)$. □

Other platforms suggested and attacks

- $\mathrm{SL}_2(\mathbb{F}_{2^n})$ [Tillich-Zémor '93], attacked by the "palindromic attack" [Grassl-Ilić-Magliveras-Steinwandt '11].
- $\mathrm{SL}_2(\mathbb{F}_p)$ [Bromberg-Shpilrain-Vdovina '17], with non-symmetric S.
- $\mathrm{SL}_2(\mathbb{F}_{2^n})$ [Ghaffari-Mostaghim '18], more secure variant.
- $\mathrm{GL}_2(\mathbb{F}_{p^n})$ [Tomkins-Nevins-Salmasian '20].

In the context of a hash function, a palindromic attack might involve creating two different messages that produce the same hash value when processed by the hash function. This means that an attacker can craft two distinct messages with the same hash, potentially leading to collisions and undermining the security properties of the hash function.

Our platform: $\mathrm{SL}_n(\mathbb{F}_p)$

Theorem (Arzhantseva-Biswas '18)

For every $n \geq 3$, there exist $A, B \in \mathrm{SL}_n(\mathbb{Z})$ such that the following holds.

1. *$\langle A, B \rangle$ is a free group of rank two,*
2. *for every large enough p, $\langle A_p, B_p \rangle = \mathrm{SL}_n(\mathbb{F}_p)$.*

Definition

We fix $n \geq 3$ and define:

- $G_p = \mathrm{SL}_n(\mathbb{F}_p)$, $S_p = \{A_p^{\pm 1}, B_p^{\pm 1}\}$,
- φ_p the hash function associated with (G_p, S_p, σ_p), for some choice of σ_p.

A message $m \in \mathbb{N}$ will be mapped to a product of the form:

$$\varphi_p(m) := A_p^{k_1} B_p^{k_2} A_p^{k_3} B_p^{k_4} \cdots A_p^{k_{l-1}} B_p^{k_l}, \quad \text{for some } \{k_\lambda\}_{1 \leq \lambda \leq l} \in \mathbb{Z}^{[1,l]}.$$

Pre-image resistance

message $\leftrightarrow$ non-backtracking walk
random message $\leftrightarrow$ non-backtracking **random** walk

Theorem (Kazhdan 60's, Alon-Benjamini-Lubetzky-Sodin '07)

Let $G = \mathrm{SL}_n(\mathbb{Z})$, with $n \geq 3$. Let $A, B \in G$ be as above. Recall $S_p := \{A_p^{\pm 1}, B_p^{\pm 1}\}$ and $G_p := \mathrm{SL}_n(\mathbb{F}_p)$. Let $N = |G_p|$. Let $\{X_\lambda\}_{\lambda \geq 0}$ be a non-backtracking random walk in $\mathrm{Cay}(G_p, S_p)$. Then, there is a constant $C > 0$ such that whenever $l \geq C \log p$, we have:

$$|\Pr(X_l = v) - 1/N| \leq 1/N^2,$$

For every $v \in G_p$.

Conclusion: the output of a random message will look random.
This is a step towards preimage resistance.

Collision resistance

Proposition

There is $c > 0$ such that every loop of $\mathrm{Cay}(G_p, S_p)$ has size at least $c \log p$ (this is optimal).

Proof.

A loop in $\mathrm{Cay}(G_p, S_p)$ can be interpreted algebraically as:

$$A_p^{k_1} B_p^{k_2} A_p^{k_3} B_p^{k_4} \cdots A_p^{k_{l-1}} B_p^{k_l} = \mathrm{id},$$

$\langle A, B \rangle$ is free $\Rightarrow \mathrm{Cay}(\langle A, B \rangle, \{A^{\pm 1}, B^{\pm 1}\})$ is a tree. So:

$$A^{k_1} B^{k_2} A^{k_3} B^{k_4} \cdots A^{k_{l-1}} B^{k_l} \neq \mathrm{id},$$

One of the entry in the matrix above has to be bigger than $p - 1$. This can happen only if the product has at least $c' \log p$ terms. □

Related mathematical problems

Factorizing problem in $\mathrm{SL}_n(\mathbb{F}_p)$.

Fact

The preimage problem for φ_p is equivalent to the following problem: Given $M \in G_p$, find $\{k_\lambda\}_{1\leq\lambda\leq l} \in \mathbb{Z}^{[1,l]}$, such that:

$$A_p^{k_1} B_p^{k_2} A_p^{k_3} B_p^{k_4} \cdots A_p^{k_{l-1}} B_p^{k_l} = M.$$

When $M = \mathrm{id}$, this is equivalent to the collision problem.

Related mathematical problems

Multivariate polynomial equations over $\mathbb{F}_p$, NP-complete [Fraenkel-Yesha '80], good for PQC [Bennett et al. '97].

Proposition

The equation

$$A_p^{k_1} B_p^{k_2} A_p^{k_3} B_p^{k_4} \cdots A_p^{k_{l-1}} B_p^{k_l} = M,$$

is equivalent to a constrained multivariate polynomial system of equations over $\mathbb{F}_p$ satisfying:

- *n^2 equations*
- *$\simeq \log p$ variables $\bar{k}_1, \ldots, \bar{k}_l \in \mathbb{F}_p$*
- *solutions has to be small: there exist $k_1, \ldots, k_l \in \mathbb{Z}$ such that:*
  - *$k_\lambda \mod p = \bar{k}_\lambda$, for every λ,*
  - *$\sum_\lambda |k_\lambda| = O(\log p)$.*

Comments and future work

- Choosing parameters: n, p.
- What means a random message? (encoding)
- Explicit estimates of constants involved?
- Develop polynomial interpretation to justify complexity?

Possible choices for $\tilde{A}$ and $\tilde{B}$ are given by:

$$\tilde{A} = \begin{pmatrix} 1 & a & 0 & 0 & \dots & 0 \\ 0 & 1 & a & 0 & \dots & 0 \\ 0 & 0 & 1 & a & \dots & 0 \\ \vdots & & & & & \vdots \\ & & & & \dots & a \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \tilde{B} = \begin{pmatrix} 1 & 0 & 0 & \dots & & 0 \\ b & 1 & 0 & \dots & & 0 \\ 0 & b & 1 & \dots & & 0 \\ 0 & 0 & b & \dots & & 0 \\ \vdots & & & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b & 1 \end{pmatrix} \in \mathrm{SL}_n(\mathbb{Z}),$$

with $a, b \geq 2$. These matrices will be crucial in the description of our hash function.

Definition 2.6. Let p be a prime, and let

$$A = \begin{pmatrix} 1 & 4 & 0 & 0 & \dots & 0 \\ 0 & 1 & 4 & 0 & \dots & 0 \\ 0 & 0 & 1 & 4 & \dots & 0 \\ \vdots & & & & & \vdots \\ & & & & \dots & 4 \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}^4, \quad B = \begin{pmatrix} 1 & 0 & 0 & \dots & & 0 \\ 2 & 1 & 0 & \dots & & 0 \\ 0 & 2 & 1 & \dots & & 0 \\ 0 & 0 & 2 & \dots & & 0 \\ \vdots & & & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & 1 \end{pmatrix}^4 \in \mathrm{SL}_n(\mathbb{F}_p),$$

Let $s(1) = A$, $s(2) = B$, $s(3) = A^{-1}$, $s(4) = B^{-1}$. We define the functions $\{s_\lambda\}_{\lambda \in [4]}$ as follows:

- $s_1(1) = B$, $s_1(2) = A^{-1}$, $s_1(3) = B^{-1}$,
- $s_2(1) = A$, $s_2(2) = A^{-1}$, $s_2(3) = B^{-1}$,
- $s_3(1) = A$, $s_3(2) = B^{-1}$, $s_3(3) = B$,
- $s_4(1) = A$, $s_4(2) = A^{-1}$, $s_4(3) = B$,

Given an input sequence $x = \{x_i\}_{i \in [1,k]} \in [3]^k$, we inductively define:

- $B_1 = s_1(x_1)$
- $B_i = s_\lambda(x_i)$, with $\lambda = s^{-1}(B_{i-1}^{-1})$, for each $k \in [2, k]$.

Then, the sequence x is hashed to the matrix:

$$\varphi(x) = B_1 \cdots B_k.$$

Thus, we obtain a hash function for every $n \geq 3$.

Example 2.7. With $n = 3$ we have:

$$A = \begin{pmatrix} 1 & 16 & 96 \\ 0 & 1 & 16 \\ 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 \\ 8 & 1 & 0 \\ 24 & 8 & 1 \end{pmatrix} \in \mathrm{SL}_3(\mathbb{F}_p),$$

For example, if we consider the sequence $x = 2232221$, following the procedure above we obtain the sequence:

- $B_1 = s_1(2) = A^{-1}$
- $B_2 = s_1(2) = A^{-1}$, where we use the map s_1 because $B_1^{-1} = s(1)$,
- $B_3 = s_1(3) = B^{-1}$, where we use the map s_1 because $B_2^{-1} = s(1)$,
- $B_4 = s_2(2) = A^{-1}$, where we use the map s_2 because $B_3^{-1} = s(2)$,
- $B_5 = s_1(2) = A^{-1}$, where we use the map s_1 because $B_4^{-1} = s(1)$,
- $B_6 = s_1(2) = A^{-1}$, where we use the map s_1 because $B_5^{-1} = s(1)$,
- $B_7 = s_1(1) = B$, where we use the map s_1 because $B_6^{-1} = s(1)$,

Finally, x is mapped to $B_1 B_2 \dots B_7$:

$$\varphi(x) = A^{-2}B^{-1}A^{-3}B = \begin{pmatrix} 694190977 & 233260720 & 29297952 \\ -38379648 & -12896255 & -1619792 \\ 1191936 & 400512 & 50305 \end{pmatrix} \in \mathrm{SL}_3(\mathbb{F}_p).$$

Semidirect Discrete Logarithm Problem (SDLP)

The Diffie-Hellman public key exchange (1976)

1. Alice and Bob agree on a public (finite) cyclic group G and a generating element g in G. We will write the group G multiplicatively.
2. Alice picks a random natural number a and sends g^a to Bob.
3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $K_A = (g^b)^a = g^{ba}$.
5. Bob computes $K_B = (g^a)^b = g^{ab}$.

Since $ab = ba$ (because $\mathbb{Z}$ is commutative), both Alice and Bob are now in possession of the same group element $K = K_A = K_B$ which can serve as the shared secret key.

Security assumptions

To recover g^{ab} from (g, g^a, g^b) is hard.

To recover a from (g, g^a) (discrete log problem) is hard.

Variations on Diffie-Hellman: why not just multiply them?

1. Alice and Bob agree on a (finite) cyclic group G and a generating element g in G. We will write the group G multiplicatively.
2. Alice picks a random natural number a and sends g^a to Bob.
3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $K_A = (g^b) \cdot (g^a) = g^{b+a}$.
5. Bob computes $K_B = (g^a) \cdot (g^b) = g^{a+b}$.

Obviously, $K_A = K_B = K$, which can serve as the shared secret key.
Drawback: anybody can obtain K the same way!

Semidirect product

Let G, H be two groups, let $Aut(G)$ be the group of automorphisms of G, and let $\rho : H \to Aut(G)$ be a homomorphism. Then the semidirect product of G and H is the set

$$\Gamma = G \rtimes_\rho H = \{(g, h) : g \in G,\ h \in H\}$$

with the group operation given by

$$(g, h)(g', h') = (g^{\rho(h')} \cdot g',\ h \cdot h').$$

Here $g^{\rho(h')}$ denotes the image of g under the automorphism $\rho(h')$.

Extensions by automorphisms

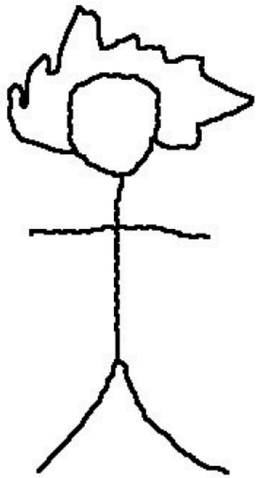
If $H = Aut(G)$, then the corresponding semidirect product is called the *holomorph* of the group G. Thus, the holomorph of G, usually denoted by $Hol(G)$, is the set of all pairs $(g,\ \phi)$, where $g \in G,\ \phi \in Aut(G)$, with the group operation given by

$$(g,\ \phi) \cdot (g',\ \phi') = (\phi'(g) \cdot g',\ \phi \cdot \phi').$$

It is often more practical to use a subgroup of $Aut(G)$ in this construction.

Also, if we want the result to be just a semigroup, not necessarily a group, we can consider the semigroup $End(G)$ instead of the group $Aut(G)$ in this construction.

Public Key-exchange Using Semidirect Products of (semi)Groups



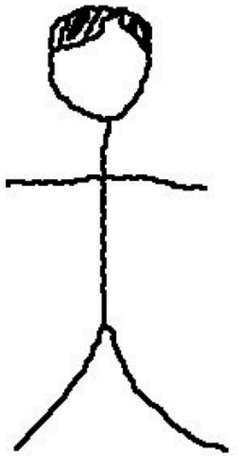
Public: G, $g \in G$, ϕ.

a, b

$(b, x) \cdot (a,\ \phi^m) =$

$(a,\ y) \cdot (b,\ \phi^n) =$

$(g,\ \phi)^{m+n}$



Private key: $n \in \mathbb{N}$

b

Private key: $m \in \mathbb{N}$

$(g, \phi)^m =$

$(\underbrace{\phi^{m-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g}_{a},\ \phi^m)$

The 15th International Conference on Post-Quantum Cryptography

PQCrypto 2024

Mathematical Institute, University of Oxford (UK)

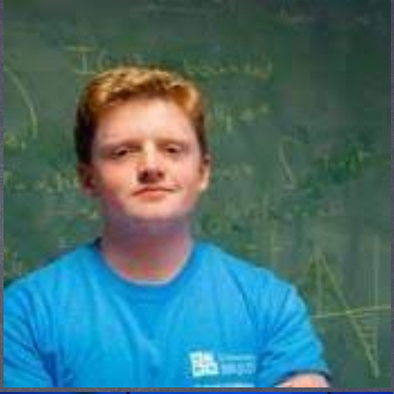
NIST

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

EVENTS 2022

Fourth PQC Standardization Conference



A Subexponential Quantum Algorithm for the Semidirect Discrete Logarithm Problem

Christopher Battarbee, University of York
Delaram Kahrobaei, Queens College, CUNY
Ludovic Perret, Sorbonne University
Siamak F. Shahandashti, University of York

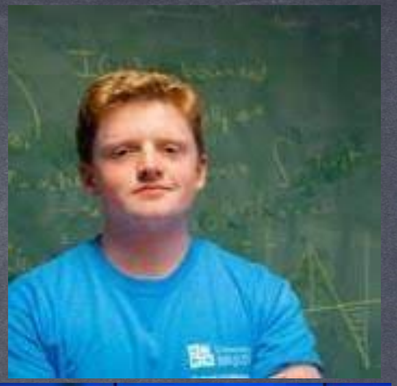
Abstract

Group-based cryptography is a relatively young family in post-quantum cryptography. In this paper we give the first dedicated security analysis of a central problem in group-based cryptography: the so-called Semidirect Product Key Exchange(SDPKE). We present a subexponential quantum algorithm for solving SDPKE. To do this we reduce SDPKE to the Abelian Hidden Shift Problem (for which there are known quantum subexponential algorithms). We stress that this does not per se constitute a break of SDPKE; rather, the purpose of the paper is to provide a connection to known problems.

PQCrypto 2023
The 14th International Conference on Post-Quantum Cryptography
August 16-18, 2023
College Park, MD, USA

SPDH-Sign: towards Efficient, Post-quantum Group-based Signatures

Christopher Battarbee, University of York
Delaram Kahrobaei, City University of New York
Ludovic Perret, Sorbonne University
Siamak F. Shahandashti, University of York



Abstract

In this paper, we present a new diverse class of post-quantum group-based Digital Signature Schemes (DSS). The approach is significantly different from previous examples of group-based digital signatures and adopts the framework of group action-based cryptography: we show that each finite group defines a group action relative to the semidirect product of the group by its automorphism group, and give security bounds on the resulting signature scheme in terms of the group-theoretic computational problem known as the Semidirect Discrete Logarithm Problem (SDLP). Crucially, we make progress towards being able to efficiently compute the novel group action, and give an example of a parameterised family of groups for which the group action can be computed for any parameters, thereby negating the need for expensive offline computation or inclusion of redundancy required in other schemes of this type.

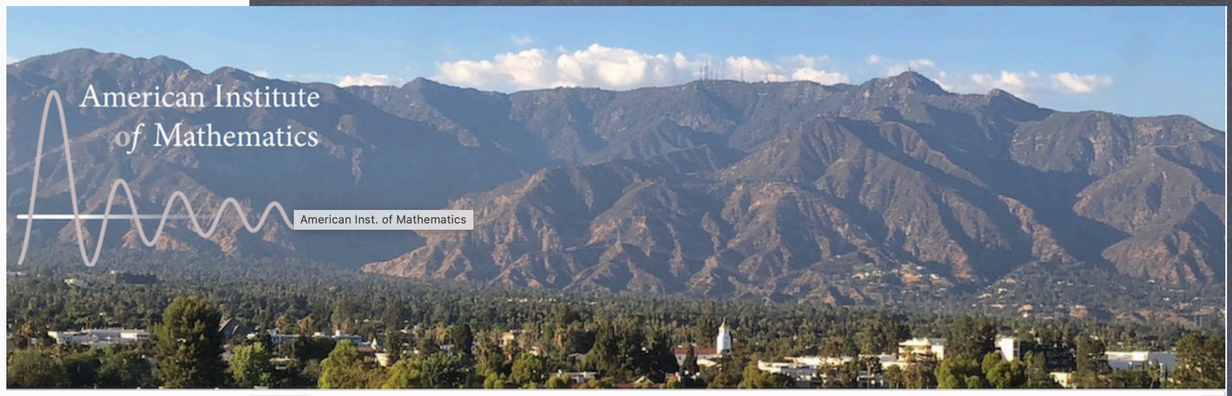
On the Semidirect Discrete Logarithm Problem in Finite Groups

Christopher Battarbee, Sorbonne University
Giacomo Borin, IBM Research Europe, University of Zurich
Ryann Cartor, Clemson University
Nadia Heninger, University of California, San Diego
David Jao, University of Waterloo
Delaram Kahrobaei, City University of New York, New York University
Laura Maddison, University of Ottawa
Edoardo Persichetti, Florida Atlantic University
Angela Robinson, National Institute of Standards and Technology
Daniel Smith-Tone, National Institute of Standards and Technology
Rainer Steinwandt, University of Alabama in Huntsville

Abstract

We present an efficient quantum algorithm for solving the semidirect discrete logarithm problem (SDLP) in any finite group. The believed hardness of the semidirect discrete logarithm problem underlies more than a decade of works constructing candidate post-quantum cryptographic algorithms from nonabelian groups. We use a series of reduction results to show that it suffices to consider SDLP in finite simple groups. We then apply the celebrated Classification of Finite Simple Groups to consider each family. The infinite families of finite simple groups admit, in a fairly general setting, linear algebraic attacks providing a reduction to the classical discrete logarithm problem. For the sporadic simple groups, we show that their inherent properties render them unsuitable for cryptographically hard SDLP instances, which we illustrate via a Baby-Step Giant-Step style attack against SDLP in the Monster Group.

Our quantum SDLP algorithm is fully constructive for all but three remaining cases that appear to be gaps in the literature on constructive recognition of groups; for these cases SDLP is no harder than finding a linear representation. We conclude that SDLP is not a suitable post-quantum hardness assumption for any choice of finite group.



Post-quantum group-based cryptography

April 29 to May 3, 2024
at the
American Institute of Mathematics, Pasadena, California
organized by
Delaram Kahrobaei and Ludovic Perret