Data Mining Over Encrypted Data Using Combinatorial Algebra Post-quantum Cryptography

Delaram Kahrobaei

THE GRADUATE CENTER CITY UNIVERSITY OF NEW YORK
CUNY The City University of New York
NEW YORK UNIVERSITY

NASA IS&T Colloquium, May 17, 2017

Motivation for Group-based Cryptography

- New one-way functions
- Shor quantum algorithm
- Provable security and quantum resistant
- Efficiency

FHE

Definition

A Fully Homomorphic Encryption (FHE) scheme is an encryption scheme which is additively and multiplicatively homomorphic. For encryption function E, and plaintexts a, b, the following holds:

$$E(a + b) = E(a) + E(b)$$

$$E(ab) = E(a)E(b)$$

It is implied that for any polynomial function F, evaluating the function on ciphertexts and decrypting the result is equivalent to evaluating the function on plaintexts, x_i.

$$E(F(x_1, x_2, \ldots, x_n)) = F(E(x_1), E(x_2), \ldots, E(x_n))$$

Black Box

FHE Scheme:

D. Kahrobaei, V. Shpilrain, *Method and apparatus for fully homomorphic encryption, private search, and private information retrieval*, US20170063526.

- Plaintext Space: Z_p, embedded into a ring R which is a direct sum of several copies of Z_p
- Ciphertext Space: $\mathcal{S}$, another direct sum of several copies of Z_p, with ideal $\mathcal{I}$ such that $\mathcal{S}/\mathcal{I} = R$
- Encryption: for $u \in R$, and $E(0) \in_R \mathcal{I}$

$$E(u) = u + E(0)$$

- Decryption: a map $\rho : \mathcal{S} \to R$ where $\rho(\mathcal{I}) = 0$.

Black Box

FHE Scheme:

D. Kahrobaei, V. Shpilrain, *Method and apparatus for fully homomorphic encryption, private search, and private information retrieval*, US20170063526.

$$Z_p \xrightarrow{\alpha} R \xrightarrow{E} \mathcal{S} \xrightarrow{\rho} R \xrightarrow{\beta} Z_p$$

Correctness of Evaluation: For $j_1, j_2, j_3 \in \mathcal{I}$ and $u, v \in R$, the scheme presented above is both additively and multiplicatively homomorphic:

$$E(u) + E(v) = u + j_1 + v + j_2 = u + v + j_3 = E(u + v)$$

$$E(u)E(v) = (u+j_1)(v+j_2) = uv+uj_2+j_1v+j_1j_2 = uv+j_3 = E(uv)$$

Black Box

FHE Scheme:

D. Kahrobaei, V. Shpilrain, *Method and apparatus for fully homomorphic encryption, private search, and private information retrieval*, US20170063526.

$$Z_p \xrightarrow{\alpha} R \xrightarrow{E} \mathcal{S} \xrightarrow{\rho} R \xrightarrow{\beta} Z_p$$

Security:

- Secure against ciphertext-only attack (COA)
- An attacker who retrieves part of an encrypted database has only a negligible probability of correctly decrypting any portion of the database.

Goals

Goal: Secure, accurate and efficient machine learning and function evaluation over encrypted data.

Private search and Third Party Search

$$P(x) = \Pi_{E(y) \in E(D)}(E(x) - E(y)) = \Pi_{y \in D} E(x - y)$$

Fully Homomorphic Machine Learning: Encrypted medical and genomic data

Collaboration with University of Michigan, Computational Medicine and Bioinformatics Department

A. Gribov, K. Horan, D. Kahrobaei, V. Shpilrain, R. Souroush, K. Najarian, *Medical diagnostic based on encrypted medical data*, Journal of Biomedical Informatics, Elsevier, accepted pending on minor revision, 1–10 (2017).

Post-Quantum Cryptography

The quantum-safe primitives under consideration all come from the following five families:

- Lattice-based primitives where the security depends on the difficulty of solving a short or close vector problem in a lattice.
- Multivariate primitives where the security depends on the difficulty of solving a system of multivariate polynomial equations.
- Code-based primitives where the security depends on the difficulty of solving a decoding problem in a linear code.
- Hash-based primitives where the security depends on the difficulty of finding collisions or preimages in cryptographic hash functions.
- Isogeny-based key primitives where the security depends on the difficulty of finding an unknown isogeny between a pair of supersingular elliptic curves.

Post-Quantum Cryptography

Definition (Superposition Attack)

A superposition attack can be applied to both classical and quantum cryptographic protocols, where we consider security if the adversary had quantum access to the primitive, i.e. quantum communication with the oracle. Therefore, the adversary can query the oracle in a superposition of states.

Presentation

Presentations are ways of defining groups as quotient of free groups.

Definition (Relation)

Let X be an alphabet. A relation over X is any pair (w, w') of reduced words over X. We usually write $w = w'$ instead of (w, w'). Note that $ww'^{-1} = 1$ (where 1 denoted the identity). Let R be the set of all relations. We define a presentation as a pair $\langle X; R \rangle$.

Definition (Group Presentation)

Let $F(X)$ be the free group on the set X. Define N to be the smallest normal subgroup of $F(X)$ containing the set $\{ww'^{-1} | (w = w') \in R\}$. Then the group defined by a presentation $\langle X; R \rangle$ is the quotient $F(X)/N$.

Decision and Search Problems

- The one-way functions that are used in group based cryptosystems are based for the most part on algorithmic **group decision and search problems**.
- We define some of the decision and search problems in combinatorial group theory, that have been used for non-commutative cryptography.
- In general all are algorithmically unsolvable.
- It is important to see how difficult they are when solvable, they are for a given platform group.

Decision, Witness and Search Problems

Definition (Decision problems)

are problems of the following nature: given a property P and an object O, find out whether or not the object O has the property P.

Definition (Witness problems)

are: given a property P and an object O with the property P, find a proof of the fact that O indeed has the property P.

Definition (Search problems)

are of the following nature: given a property P and an object O with the property P, find something material establishing the property P.

Word Problems

Definition (Word Decision Problem)

Given a finitely presented group G does there exist an algorithm to decide whether or not a word in the generators is the trivial word?

Definition (Word Search Problem)

Given a finitely presented group G and a w which presents the identity, does there exist an algorithm to find an expression of w as a product of words of the form ${f_i}^{-1} r_i f_i$ where r_i is a relator of the group G and f_i is a word in the ambient free group.

Conjugacy Problems

Definition (Decision Conjugacy Problem)

Given a group G with a finite presentation, does there exist an algorithm to decide whether or not an arbitrary pair of words u and v in the generators of G are conjugate? That is, is there an $x \in G$ such that $x^{-1}ux = v$?

Definition (Conjugator Search Problem)

Let G be a finitely presented group. Given two conjugate words u and v , is there an algorithm to find a z such that $z^{-1}uz = v$?

The problems are not independent, for example in a finitely presented group G the word problem of G is Turing reducible to the conjugacy problem of G. That is, $WP(G) \leq_T CP(G)$.

Groups with Solvable Word Problem

Example

If G is a finite group given by a multiplication table presentation, it is easy to describe algorithms for solving Decision and Search Word Problems and Conjugacy problems.

Example

If G is a finitely generated abelian group, then the word and conjugacy problems for G are solvable.

Example

If $F = \langle x_1, \cdots, x_n; \rangle$ is a finitely generated free group: WP(F) is solved by freely reducing. CP(F) is also solvable.

Example

The word problem for nilpotent group is solvable.

Example

Let G be a polycyclic group. Using the fact that every word in G has a normal form, we can conclude that G has solvable word problem.

Theorem

There exists a solvable group of class 3 *with unsolvable word problem.*

Theorem

(Novikov-Boone) There exists a finitely presented group whose word problem is recursively unsolvable.

Theorem

Conjugacy search problem is always solvable.

Growth Rate of a Group

Let G be a finitely generated group. The growth function

$$\gamma : \mathbb{N} \longrightarrow \mathbb{R}$$

is defined by

$$\gamma(n) = \#\{w \in G : l(w) \leq n\}$$

where $l(w)$ is the length of w as a word in the generators of G. If we use normal forms to represent group elements, then each element has a unique representation, and there is an obvious relation between the growth function of a group and the key space that the group provides.

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.

A Non-Abelian Diffie-Hellman Key Exchange

Non-Abelian Diffie-Hellman Key Exchange- Ko,Lee et al.

Let G be a finitely presented group such that finding the normal form of every element is efficient and solving the Conjugacy search Problem for G is hard, $A, B \leq G$ with $[A, B] = 1$.
The groups G, A, B and $g \in G$ are public.
Denote $h^{-1}gh = g^h$ for $g, h \in G$.

- Alice chooses random $a \in A$ and sends g^a to Bob.
- Bob chooses random $b \in B$ and sends g^b to Alice.
- The shared key is g^{ab} which is computable for both Bob and Alice since $[A, B] = 1$.

In order for an adversary to obtain the shared key based on the given information, he or she must solve the search conjugacy problem.

Platform Group Requirements

Properties of platform group for Non-abelian Diffie-Hellman

- Finitely Presented
- Efficiently computable normal form
- Conjugacy search problem has exponential time complexity
- Exponential growth rate (for a large key space)
- Resistant against Length Based Attack (A Heuristic Algorithm to Solve the CSP) or other existing attacks.

Polycyclic Groups (Proposed by Eick, K. 2004)

- Polycyclic Groups are finitely presented.
- Efficiently computable normal form
- The non-virtually nilpotent polycyclic groups have exponential growth rate (Milnor and Wolf 1968)
- (Garber, K, Lam [J. Math. Crypt. 2015]) Certain classes of polycyclic groups are secure against length based attacks.
- (Gryak, K., Martinez-Perez 2016) There are classes of finitely presented metabelian groups in which the conjugacy search problem is at most exponential time. In the case of generalized metabelian Baumslag-Solitar Groups reduces to Discrete Log Problem.

Generalized Metabelian BS Groups and DLP

For generalized metabelian Baumslag-Solitar groups of the form:

$$G = \langle q_1, q_2, b | b^{q_1} = b^{m_1}, b^{q_2} = b^{m_2}, [q_1, q_2] = 1 \rangle ,$$

the conjugacy search problem reduces to the discrete logarithm problem.

Proposed Platforms

- Braid groups (Ko-Lee) 2000: Complexity of Conjugacy Search Problem
- Polycyclic Groups (Eick, K.-2004): Complexity of Conjugacy Search Problem
- Linear Groups (Baumslag, Fine, Xu, 2004) : Complexity of finding generators of subgroups
- Right Angled Artin Groups (Flores, K. 2016) : Authentication Scheme: Subgroup Isomorphism problem (unsolvability results by Bridson), Group Homomorphism Problem (NP-Complete), Secret sharing: Linear time complexity Word Problem
- Hyperbolic Groups (Chatterji, K., Lu 2016): Subgroup distortion.
- Free nilpotent p-groups (K., Shpilrain 2016): Semidirect product public key

Proposed Platforms

- Semigroup of Matrices over Group Rings (K., Koupparis, Shpilrain, 2013) Discrete Log Type
- Small Cancellation Groups (Habeeb, K., Shpilrain 2012)
- Free Metabelian Groups (Shpilrain, Zapata, 2006) Subgroup Membership Search Problem, (Habeeb, K., Shpilrain 2012) : Complexity of Endomorphism Search problem.
- Thompson Groups (Shpilrain, Ushakov, 2005): Complexity of Decomposition Search Problem
- Grigorchuk Groups (Petrides 2003)
- Groups of Matrices (Grigoriev, Ponameranco, 2004) Homomorphic Encryption

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

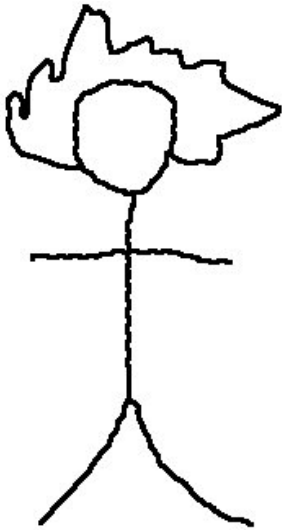
If $H = Aut(G)$, then the corresponding semidirect product is called the *holomorph* of the group G. Thus, the holomorph of G, usually denoted by $Hol(G)$, is the set of all pairs $(g,\ \phi)$, where $g \in G,\ \phi \in Aut(G)$, with the group operation given by

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

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

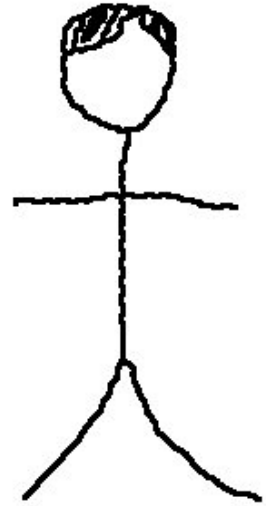
Also, if we want the result to be just a semigroup, not necessarily a group, we can consider the semigroup $End(G)$ instead of the group $Aut(G)$ in this construction.

Public Key-Exchange Using Semidirect Product of 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: $m \in \mathbb{N}$

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

M. Habeeb, D. Kahrobaei, C. Koupparis, and V. Shpilrain, *Public key exchange using semidirect product of (semi)groups*, in: ACNS 2013, Applied Cryptography and Network Security, LNCS **7954** (2013), 475–486.

D. Kahrobaei, V. Shpilrain, *Invited Paper: Using semidirect product of (semi)groups in public key cryptography*, Computability in Europe 2016, LNCS **9709**,132–141 (2016).

Key exchange using extensions by automorphisms (Habeeb-K.-Koupparis-Shpilrain)

- Let G be a group (or a semigroup).
- An element $g \in G$ is chosen and made public as well as an arbitrary automorphism (or an endomorphism) ϕ of G.
- Bob chooses a private $n \in \mathbb{N}$.
- While Alice chooses a private $m \in \mathbb{N}$.
- Both Alice and Bob are going to work with elements of the form (g, ϕ^k), where $g \in G,\ k \in \mathbb{N}$.

Using semidirect product (cont.)

1. Alice computes

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

and sends **only the first component** of this pair to Bob. Thus, she sends to Bob **only** the element

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

of the group G.

2. Bob computes

$$(g, \phi)^n = (\phi^{n-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g,\ \phi^n)$$

and sends **only the first component** of this pair to Alice:

$$b = \phi^{n-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g.$$

Using semidirect product (cont.)

3. Alice computes

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

Her key is now

$$K_A = \phi^m(b) \cdot a.$$

Note that she does not actually "compute" $x \cdot \phi^m$ because she does not know the automorphism x; recall that it was not transmitted to her. But she does not need it to compute K_A.

Using semidirect product (cont.)

4. Bob computes

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

His key is now

$$K_B = \phi^n(a) \cdot b.$$

Again, Bob does not actually "compute" $y \cdot \phi^n$ because he does not know the automorphism y.

5. Since

$$(b, x) \cdot (a,\ \phi^m) = (a,\ y) \cdot (b,\ \phi^n) = (g,\ \phi)^{m+n},$$

we should have $K_A = K_B = K$, the shared secret key.

Special case: Diffie-Hellman

$G = \mathbb{Z}_p^*$
$\phi(g) = g^k$ for all $g \in G$ and a fixed k, $1 < k < p-1$, where k is relatively prime to $p-1$.

Then $(g, \phi)^m = (\phi^{m-1}(g) \cdots \phi(g) \cdot \phi^2(g) \cdot g,\ \phi^m)$.
The first component is equal to $g^{k^{m-1}+\ldots+k+1} = g^{\frac{k^m-1}{k-1}}$.
The shared key $K = g^{\frac{k^{m+n}-1}{k-1}}$.

Special case: Diffie-Hellman

"The Diffie-Hellman type problem" would be to recover the shared key

$$K = g^{\frac{k^{m+n}-1}{k-1}}$$

from the triple

$$(g,\ g^{\frac{k^m-1}{k-1}},\ g^{\frac{k^n-1}{k-1}}).$$

Since g and k are public, this is equivalent to recovering $g^{k^{m+n}}$ from the triple $(g,\ g^{k^m},\ g^{k^n})$, i.e., this is exactly the standard Diffie-Hellman problem.

Group ring

Definition (Group ring)

Let G be a group written multiplicatively and let R be any commutative ring with nonzero unity. The group ring $R[G]$ is defined to be the set of all formal sums

$$\sum_{g_i \in G} r_i g_i$$

where $r_i \in R$, and all but a finite number of r_i are zero.

We define the sum of two elements in RG by

$$\left(\sum_{g_i \in G} a_i g_i\right) + \left(\sum_{g_i \in G} b_i g_i\right) = \sum_{g_i \in G} (a_i + b_i) g_i.$$

Note that $(a_i + b_i) = 0$ for all but a finite number of i, hence the above sum is in $R[G]$. Thus $(R[G], +)$ is an abelian group. Multiplication of two elements of $R[G]$ is defined by the use of the multiplications in G and R as follows:

$$\left(\sum_{g_i \in G} a_i g_i\right)\left(\sum_{g_i \in G} b_i g_i\right) = \sum_{g_i \in G} \left(\sum_{g_j g_k = g_i} a_j b_k\right) g_i.$$

Platform: matrices over group rings

We use the semigroup of 3×3 matrices over the group ring $\mathbb{Z}_7[A_5]$, where A_5 is the alternating group on 5 elements.
Then the public key consists of two matrices: the (invertible) conjugating matrix H and a (non-invertible) matrix M. The shared secret key then is:

$$K = H^{-(m+n)}(HM)^{m+n}.$$

Here we use an extension of the semigroup G by an inner automorphism φ_H, which is conjugation by a matrix $H \in GL_3(\mathbb{Z}_7[A_5])$. Thus, for any matrix $M \in G$ and for any integer $k \geq 1$, we have

$$\varphi_H(M) = H^{-1}MH;\ \varphi_H^k(M) = H^{-k}MH^k.$$

1. Alice and Bob agree on public matrices $M \in G$ and $H \in GL_3(\mathbb{Z}_7[A_5])$. Alice selects a private positive integer m, and Bob selects a private positive integer n.

2. Alice computes $(M, \varphi_H)^m = (H^{-m+1}MH^{m-1} \cdots H^{-2}MH^2 \cdot H^{-1}MH \cdot M,\ \varphi_H^m)$ and sends **only the first component** of this pair to Bob. Thus, she sends to Bob **only** the matrix

$$A = H^{-m+1}MH^{m-1} \cdots H^{-2}MH^2 \cdot H^{-1}MH \cdot M = H^{-m}(HM)^m.$$

3. Bob computes
$(M, \varphi_H)^n = (H^{-n+1}MH^{n-1} \dots H^{-2}MH^2 \cdot H^{-1}MH \cdot M,\ \varphi_H^n)$
and sends **only the first component** of this pair to Alice.
Thus, he sends to Alice **only** the matrix

$$B = H^{-n+1}MH^{n-1} \dots H^{-2}MH^2 \cdot H^{-1}MH \cdot M = H^{-n}(HM)^n.$$

4. Alice computes $(B, x) \cdot (A,\ \varphi_H^m) = (\varphi_H^m(B) \cdot A,\ x \cdot \varphi_H^m)$. Her key is now $K_{Alice} = \varphi_H^m(B) \cdot A = H^{-(m+n)}(HM)^{m+n}$. Note that she does not actually "compute" $x \cdot \varphi_H^m$ because she does not know the automorphism $x = \varphi_H^n$; recall that it was not transmitted to her. But she does not need it to compute K_{Alice}.

5. Bob computes $(A, y) \cdot (B,\ \varphi_H^n) = (\varphi_H^n(A) \cdot B,\ y \cdot \varphi_H^n)$. His key is now $K_{Bob} = \varphi_H^n(A) \cdot B$. Again, Bob does not actually "compute" $y \cdot \varphi_H^n$ because he does not know the automorphism $y = \varphi_H^m$.

6. Since $(B, x) \cdot (A,\ \varphi_H^m) = (A,\ y) \cdot (B,\ \varphi_H^n) = (M,\ \varphi_H)^{m+n}$, we should have $K_{Alice} = K_{Bob} = K$, the shared secret key.

Security assumptions

To recover $H^{-(m+n)}(HM)^{m+n}$ from $(M,\ H,\ H^{-m}(HM)^m,\ H^{-n}(HM)^n)$ is hard.

To recover m from $H^{-m}(HM)^m$ is hard.

Nilpotent groups and p-groups

Definition

First we recall that a *free group* F_r on $x_1, \ldots, x_r$ is the set of *reduced words* in the alphabet $\{x_1, \ldots, x_r, x_1^{-1}, \ldots, x_r^{-1}\}$.

- It is a fact that every group that can be generated by r elements is the factor group of F_r by an appropriate normal subgroup. We are now going to define two special normal subgroups of F_r.
- The normal subgroup F_r^p is generated (as a group) by all elements of the form g^p, $g \in F_r$. In the factor group F_r/F_r^p every nontrivial element therefore has order p (if p is a prime).

Nilpotent groups and p-groups (cont.)

- The other normal subgroup that we need is somewhat less straightforward to define. Let $[a, b]$ denote $a^{-1}b^{-1}ab$. Then, inductively, let $[y_1, \ldots, y_{c+1}]$ denote $[[y_1, \ldots, y_c], y_{c+1}]$. For a group G, denote by $\gamma_c(G)$ the (normal) subgroup of G generated (as a group) by all elements of the form $[y_1, \ldots, y_c]$. If $\gamma_{c+1}(G) = \{1\}$, we say that the group G is nilpotent of nilpotency class c.
- The factor group $F_r/\gamma_{c+1}(F_r)$ is called *the free nilpotent group* of nilpotency class c. This group is infinite.

Free nilpotent p-group

- The group $G = F_r/F_r^{p^2} \cdot \gamma_{c+1}(F_r)$ is what we suggest to use as the platform for the key exchange protocol.
- This group, being a nilpotent p-group, is finite. Its order depends on p, c, and r. For efficiency reasons, it seems better to keep c and r fairly small (in particular, we suggest $c = 2$ or 3), while p should be large enough to make the dimension of linear representations of G so large that a linear algebra attack would be infeasible.
- The minimal faithful representation of a finite p-group as a group of matrices over a finite field of characteristic p is in this case of dimension $1 + p$. Thus, if p is, say, a 100-bit number, a linear algebra attack is already infeasible.

Thank You!