Cryptanalysis via Algebraic Spans

Adi Ben-Zvi, Arkadius Kalka, and Boaz Tsaban

Bar-Ilan University

Crypto 2018

PKC foundations are mainly abelian (and quantum insecure)

PKC foundations are mainly abelian (and quantum insecure)

DLP in finite fields (1976); Factorization (RSA, 1978).

Poor performance vs security tradeoff; no long-term security.

Subexp algorithms for DLP in some elliptic curves.

Quantum computers break them all.

PKC foundations are mainly abelian (and quantum insecure)

DLP in finite fields (1976); Factorization (RSA, 1978).

Poor performance vs security tradeoff; no long-term security.

Subexp algorithms for DLP in some elliptic curves.

Quantum computers break them all.

Options: (0) Abelian (DLP/RSA); (1) Lattices; (2) **nonabelian** groups/structures.

PKC foundations are mainly abelian (and quantum insecure)

DLP in finite fields (1976); Factorization (RSA, 1978).

Poor performance vs security tradeoff; no long-term security.

Subexp algorithms for DLP in some elliptic curves.

Quantum computers break them all.

Options: (0) Abelian (DLP/RSA); (1) Lattices; (2) **nonabelian** groups/structures.

The nonablian option must be explored.

In particular, we need general cryptanalytic tools for nonabelian crypto.

PKC foundations are mainly abelian (and quantum insecure)

DLP in finite fields (1976); Factorization (RSA, 1978).

Poor performance vs security tradeoff; no long-term security.

Subexp algorithms for DLP in some elliptic curves.

Quantum computers break them all.

Options: (0) Abelian (DLP/RSA); (1) Lattices; (2) **nonabelian** groups/structures.

The nonablian option must be explored.

In particular, we need general cryptanalytic tools for nonabelian crypto.

Here: Algebraic Span Cryptanalysis.

Conojugation in nonabelian groups

Conojugation in nonabelian groups

For $a, c \in G$ (nonabelian group),

$$a^c := c^{-1}ac$$

(conjugation).

Conojugation in nonabelian groups

For $a, c \in G$ (nonabelian group),

$$a^c := c^{-1}ac$$

(conjugation).

Conjugation is an isomorphism:

$$(a^{-1})^c = (a^c)^{-1}$$
$$(ab)^c = a^c \cdot b^c.$$

For a word $v(x_1, \ldots, x_k)$ in the variables $x_1, \ldots, x_k$ (e.g., $x_7 x_3^{-1} x_5$):

$$v(a_1^c, \ldots, a_k^c) = v(a_1, \ldots, a_k)^c.$$

Commutator KE (Anshel–Anshel–Goldfeld 1999)

Commutator KE (Anshel–Anshel–Goldfeld 1999)

Alice	**Public**	**Bob**
$v(x_1, \ldots, x_k)$	$\langle a_1, \ldots, a_k \rangle \leq G$	$w(x_1, \ldots, x_k)$
$a = v(a_1, \ldots, a_k)$	$\langle b_1, \ldots, b_k \rangle \leq G$	$b = w(b_1, \ldots, b_k)$

$$\xrightarrow{\boxed{{b_1}^a}, \ldots, \boxed{{b_k}^a}}$$

$$\xleftarrow{\boxed{{a_1}^b}, \ldots, \boxed{{a_k}^b}}$$

$a^{-1} v(\boxed{a_1^b}, \ldots, \boxed{a_k^b})$		$w(\boxed{b_1^a}, \ldots, \boxed{b_k^a})^{-1} b$

Commutator KE (Anshel–Anshel–Goldfeld 1999)

Alice	**Public**	**Bob**
$v(x_1, \ldots, x_k)$	$\langle a_1, \ldots, a_k \rangle \leq G$	$w(x_1, \ldots, x_k)$
$a = v(a_1, \ldots, a_k)$	$\langle b_1, \ldots, b_k \rangle \leq G$	$b = w(b_1, \ldots, b_k)$

$$\xrightarrow{\boxed{{b_1}^a}, \ldots, \boxed{{b_k}^a}}$$

$$\xleftarrow{\boxed{{a_1}^b}, \ldots, \boxed{{a_k}^b}}$$

$$a^{-1} v(\boxed{a_1^b}, \ldots, \boxed{a_k^b}) \qquad\qquad w(\boxed{b_1^a}, \ldots, \boxed{b_k^a})^{-1} b$$

$$a^{-1} v(a_1^b, \ldots, a_k^b) = a^{-1} a^b = a^{-1} b^{-1} ab = (b^a)^{-1} b = w(b_1^a, \ldots, b_k^a)^{-1} b$$

Linear equations from conjugations

Linear equations from conjugations

Assume $G \leq \mathrm{GL}_n(\mathbb{F})$ (matrix representations).

Linear equations from conjugations

Assume $G \leq \mathrm{GL}_n(\mathbb{F})$ (matrix representations).

Given $c = \boxed{b^a}$ ($a, b \in G$):

$$\boxed{b^a} = a^{-1}ba$$
$$a \cdot \boxed{b^a} = ba$$

Linear equations in the entries of the matrix a.

Linear equations from conjugations

Assume $G \leq \mathrm{GL}_n(\mathbb{F})$ (matrix representations).

Given $c = \boxed{b^a}$ ($a, b \in G$):

$$\begin{aligned} \boxed{b^a} &= a^{-1} b a \\ a \cdot \boxed{b^a} &= b a \end{aligned}$$

Linear equations in the entries of the matrix a.

A solution $\tilde{a}$ is invertible w.h.p. (Schwartz–Zippel).

$$\begin{aligned} \tilde{a} \cdot \boxed{b^a} &= b \tilde{a} \\ \boxed{b^a} &= \tilde{a}^{-1} b \tilde{a} \\ \boxed{b^a} &= b^{\tilde{a}} \end{aligned}$$

Algebraic spans

Algebraic spans

$G \leq \mathrm{GL}_n(\mathbb{F})$, $a, b \in G$.

Can find $\tilde{a}$ with $\boxed{b^a} = b^{\tilde{a}}$ by linear equations.

Algebraic spans

$G \leq \mathrm{GL}_n(\mathbb{F})$, $a, b \in G$.

Can find $\tilde{a}$ with $\boxed{b^a} = b^{\tilde{a}}$ by linear equations.

$\tilde{a} \notin G$!

We can force

$$\tilde{a} \in \mathrm{Alg}(G) = \mathrm{span}_{\mathbb{F}}(G) \subseteq \mathrm{M}_n(\mathbb{F}),$$

the algebra generated by G (because that's a vector space.)

Algebraic spans

$G \leq \mathrm{GL}_n(\mathbb{F})$, $a, b \in G$.

Can find $\tilde{a}$ with $\boxed{b^a} = b^{\tilde{a}}$ by linear equations.

$\tilde{a} \notin G$!

We can force

$$\tilde{a} \in \mathrm{Alg}(G) = \mathrm{span}_{\mathbb{F}}(G) \subseteq \mathrm{M}_n(\mathbb{F}),$$

the algebra generated by G (because that's a vector space.)

For $G = \langle g_1, \ldots, g_k \rangle \leq \mathrm{GL}_n(\mathbb{F})$, finding a basis for $\mathrm{Alg}(G)$ by repeated multiplication by generators and Gauss elimination is $O(kn^6)$.

Algebraic Span Cryptanalysis

Algebraic Span Cryptanalysis

$G_1, \dots, G_k \leq \mathrm{GL}_n(\mathbb{F})$; $g_1 \in G_1, \dots, g_k \in G_k$.

Given: linear equations on the entries of $g_1, \dots, g_k$.

Need to find $f(g_1, \dots, g_k)$.

Algebraic Span Cryptanalysis

$G_1, \ldots, G_k \leq \mathrm{GL}_n(\mathbb{F})$; $g_1 \in G_1, \ldots, g_k \in G_k$.

Given: linear equations on the entries of $g_1, \ldots, g_k$.

Need to find $f(g_1, \ldots, g_k)$.

Instead of solving subject to

$$g_1 \in G_1, \ldots, g_k \in G_k,$$

(infeasible!) solve subject to the linear constraints

$$g_1 \in \mathrm{Alg}(G_1), \ldots, g_k \in \mathrm{Alg}(G_k).$$

Pray (or prove) that **every** solution $\tilde{g}_1, \ldots, \tilde{g}_k$ satisfies

$$f(\tilde{g}_1, \ldots, \tilde{g}_k) = f(g_1, \ldots, g_k).$$

Application: Commutator KEP

Application: Commutator KEP

$a \in \langle a_1, \ldots, a_k \rangle, b \in \langle b_1, \ldots, b_k \rangle \leq G \leq \mathrm{GL}_n(\mathbb{F})$.

Need: $({b_1}^{a}, \ldots, {b_k}^{a}, {a_1}^{b}, \ldots, {a_k}^{b}) \mapsto a^{-1} b^{-1} a b$.

Application: Commutator KEP

$a \in \langle a_1, \dots, a_k \rangle$, $b \in \langle b_1, \dots, b_k \rangle \leq G \leq \mathrm{GL}_n(\mathbb{F})$.

Need: $({b_1}^a, \dots, {b_k}^a, {a_1}^b, \dots, {a_k}^b) \mapsto a^{-1}b^{-1}ab$.

Solving linear equations, we obtain $\tilde{a} \in \mathrm{Alg}(a_1, \dots, a_k)$, $\tilde{b} \in \mathrm{Alg}(b_1, \dots, b_k)$ with

$$\begin{array}{rcl} {b_1}^{\tilde{a}} & = & \boxed{{b_1}^a} \\ & \vdots & \\ {b_k}^{\tilde{a}} & = & \boxed{{b_k}^a} \end{array} \quad ; \quad \begin{array}{rcl} {a_1}^{\tilde{b}} & = & \boxed{{a_1}^b} \\ & \vdots & \\ {a_k}^{\tilde{b}} & = & \boxed{{a_k}^b} \end{array}$$

Since $\tilde{a} \in \mathrm{Alg}(a_1, \dots, a_k)$, $\tilde{a}^{\tilde{b}} = \tilde{a}^b$. Similarly, $b^{\tilde{a}} = b^a$.

$\tilde{a}^{-1}\tilde{b}^{-1}\tilde{a}\tilde{b} = \tilde{a}^{-1}\tilde{a}^{\tilde{b}} = \tilde{a}^{-1}\tilde{a}^b = \tilde{a}^{-1}b^{-1}\tilde{a}b = (b^{\tilde{a}})^{-1}b = (b^a)^{-1}b = a^{-1}b^{-1}ab$!

Triple Decomposition KE (Kurt 2005)

Alice **Public** **Bob**

$$a, a_1, a_2, x_1, x_2 \qquad \begin{array}{ccccccc} A & A_1 & A_2 & X_1 & X_2 & & \\ & | & | & | & | & & \leq G \\ & Y_1 & Y_2 & B_1 & B_2 & B & \end{array} \qquad y_1, y_2, b_1, b_2, b$$

$$\boxed{ax_1}, \boxed{x_1^{-1} a_1 x_2}, \boxed{x_2^{-1} a_2} \longrightarrow$$

$$\longleftarrow \boxed{b_1 y_1}, \boxed{y_1^{-1} b_2 y_2}, \boxed{y_2^{-1} b}$$

$$a \boxed{b_1 y_1} a_1 \boxed{y_1^{-1} b_2 y_2} a_2 \boxed{y_2^{-1} b} = \underbrace{a b_1 a_1 b_2 a_2 b}_{K} = \boxed{ax_1} b_1 \boxed{x_1^{-1} a_1 x_2} b_2 \boxed{x_2^{-1} a_2} b$$

The triple products do not provide linear equations! (And without them we fail!)

Cryptanalysis of Triple Dec KE

$$\mathrm{Alg}(B_1)y_1 = \mathrm{Alg}(B_1) \cdot \boxed{b_1 y_1}$$

$$\mathrm{Alg}(B_2 \cup Y_2)y_1 = \mathrm{Alg}(B_2 \cup Y_2) \cdot y_2^{-1} b_2^{-1} y_1 = \mathrm{Alg}(B_2 \cup Y_2) \cdot \boxed{y_1^{-1} b_2 y_2}^{-1}$$

$$\mathrm{Alg}(A_2)x_2 = \mathrm{Alg}(A_2) \cdot a_2^{-1} x_2 = \mathrm{Alg}(A_2) \cdot \boxed{x_2^{-1} a_2}^{-1}$$

$$\mathrm{Alg}(A_1 \cup X_1)x_2 = \mathrm{Alg}(A_1 \cup X_1) \cdot \boxed{x_1^{-1} a_1 x_2}$$

Pick invertible

$$\tilde{y}_1 \in \mathrm{Alg}(Y_1) \cap \mathrm{Alg}(B_1)y_1 \cap \mathrm{Alg}(B_2 \cup Y_2)y_1;$$

$$\tilde{x}_2 \in \mathrm{Alg}(X_2) \cap \mathrm{Alg}(A_2)x_2 \cap \mathrm{Alg}(A_1 \cup X_1)x_2.$$

$$\boxed{a x_1} \cdot \boxed{b_1 y_1} \cdot \tilde{y}_1^{-1} \cdot \boxed{x_1^{-1} a_1 x_2} \cdot \tilde{x}_2^{-1} \cdot \tilde{y}_1 \cdot \boxed{y_1^{-1} b_2 y_2} \cdot \tilde{x}_2 \cdot \boxed{x_2^{-1} a_2} \cdot \boxed{y_2^{-1} b}$$

Gives (intricate proof) $a b_1 a_1 b_2 a_2 b = K$! (Alternatively, could check empirically.)

Final comments

Final comments

Method also applies to: Nonabelian Diffie–Hellman (Ko–Lee–Cheon–Han–Kang–Park 2000), Centralizer KE (Shpilrain–Ushakov 2006), and some more.

Final comments

Method also applies to: Nonabelian Diffie–Hellman (Ko–Lee–Cheon–Han–Kang–Park 2000), Centralizer KE (Shpilrain–Ushakov 2006), and some more.

Not the end of nonabelian cryptography:

1. Additional nonabelian proposals
   (Dehornoy et al., Kalka, . . .).
2. Additional problems (CSP, Multiple CSP,. . .) to build upon.
3. Groups with no small-dim representations.
4. The application of this method keeps getting harder as new systems emerge (cf. recent cryptanalysis of Algebraic Eraser).

Final comments

Method also applies to: Nonabelian Diffie–Hellman (Ko–Lee–Cheon–Han–Kang–Park 2000), Centralizer KE (Shpilrain–Ushakov 2006), and some more.

Not the end of nonabelian cryptography:

1. Additional nonabelian proposals (Dehornoy et al., Kalka, ...).
2. Additional problems (CSP, Multiple CSP,...) to build upon.
3. Groups with no small-dim representations.
4. The application of this method keeps getting harder as new systems emerge (cf. recent cryptanalysis of Algebraic Eraser).

THANK YOU!