A SAT-based approach for index calculus on binary elliptic curves

Monika Trimoska Sorina Ionica Gilles Dequen

MIS, Université de Picardie Jules Verne

AGCCT Conference
1 June 2021

Discrete log problem

Defining discrete log problem

Given a finite cyclic group $(G, +)$ of order N and two elements $g, h \in G$, find $x \in \mathbb{Z}$ such that

$$h = x \cdot g.$$

- Generic attacks - Pollard rho, Baby-step Giant-step, Kangaroo
- Index calculus attack : subexponential in $(\mathbb{Z}/p\mathbb{Z})*$.

Index calculus on binary elliptic curves

Let $\mathbb{F}_{2^n}$ be a finite field and E be an elliptic curve defined by

$$E : y^2 + xy = x^3 + ax^2 + b$$

with $a, b \in \mathbb{F}_{2^n}$ and n prime.

Index calculus on binary elliptic curves

Let $\mathbb{F}_{2^n}$ be a finite field and E be an elliptic curve defined by

$$E : y^2 + xy = x^3 + ax^2 + b$$

with $a, b \in \mathbb{F}_{2^n}$ and n prime.

1. Choice of an appropriate factor base $\mathcal{B}$
2. Point decomposition phase

   Find $P_1, \ldots, P_{m-1} \in \mathcal{B}$, such that, for $R \in E(\mathbb{F}_{2^n})$

   $$R = P_1 + \ldots + P_{m-1}$$

3. Linear algebra

Point Decomposition Problem (PDP)

Semaev's summation polynomials (2004)

$$S_2(X_1, X_2) = X_1 + X_2,$$
$$S_3(X_1, X_2, X_3) = X_1^2X_2^2 + X_1^2X_3^2 + X_1X_2X_3 + X_2^2X_3^2 + b,$$

For $m \geq 4$

$$S_m(X_1, \ldots, X_m) =$$
$$Res_X(S_{m-k}(X_1, \ldots, X_{m-k-1}, X), S_{k+2}(X_{m-k}, \ldots, X_m, X))$$

Point Decomposition Problem (PDP)

Semaev's summation polynomials (2004)

$$S_2(X_1, X_2) = X_1 + X_2,$$
$$S_3(X_1, X_2, X_3) = X_1^2X_2^2 + X_1^2X_3^2 + X_1X_2X_3 + X_2^2X_3^2 + b,$$

For $m \geq 4$

$$S_m(X_1, \ldots, X_m) =$$
$$Res_X(S_{m-k}(X_1, \ldots, X_{m-k-1}, X), S_{k+2}(X_{m-k}, \ldots, X_m, X))$$

Reducing the PDP to the problem of finding the roots of S_m

For $R, P_1, \ldots, P_{m-1} \in E(\mathbb{F}_{2^n})$

$$R + P_1 + \ldots + P_{m-1} = \mathcal{O} \iff S_m(\mathbf{x}_R, \mathbf{x}_{P_1}, \ldots, \mathbf{x}_{P_{m-1}}) = 0$$

Using symmetries

Gaudry (2008)

Symmetrization

Rewrite S_m in terms of the elementary symmetric polynomials

$$
\begin{aligned}
\mathbf{e}_1 &= \sum_{1 \le i_1 \le m} X_{i_1}, \\
\mathbf{e}_2 &= \sum_{1 \le i_1, i_2 \le m} X_{i_1} X_{i_2}, \\
&\dots \\
\mathbf{e}_m &= \prod_{1 \le i \le m} X_i.
\end{aligned}
$$

PDP algebraic model

Yun-Ju *et al.* (2013)

Factor base for elliptic curves defined over $\mathbb{F}_{2^n}$, with n prime

An l-dimensional vector subspace V of $\mathbb{F}_{2^n}/\mathbb{F}_2$. When $l \sim \frac{n}{m}$ the system has a reasonable chance to have a solution.

PDP algebraic model

Yun-Ju *et al.* (2013)

Factor base for elliptic curves defined over $\mathbb{F}_{2^n}$, with n prime

An l-dimensional vector subspace V of $\mathbb{F}_{2^n}/\mathbb{F}_2$. When $l \sim \frac{n}{m}$ the system has a reasonable chance to have a solution.

Let t be a root of a defining polynomial of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$.

X_i-variables

$$X_1 = c_{1,0} + \ldots + c_{1,l-1}t^{l-1}$$
$$X_2 = c_{2,0} + \ldots + c_{2,l-1}t^{l-1}$$
$$\ldots$$
$$X_m = c_{m,0} + \ldots + c_{m,l-1}t^{l-1}$$

e_i-variables

$$e_1 = d_{1,0} + \ldots + d_{1,l-1}t^{l-1}$$
$$e_2 = d_{2,0} + \ldots + d_{2,2l-2}t^{2l-2}$$
$$\ldots$$
$$e_m = d_{m,0} + \ldots + d_{m,m(l-1)}t^{m(l-1)}$$

PDP algebraic model

Two sets of equations

- Equations defining symmetric polynomials

$$
\begin{aligned}
d_{1,0} &= c_{1,0} + \ldots + c_{m,0} \\
d_{1,1} &= c_{1,1} + \ldots + c_{m,1} \\
\ldots \\
d_{m,m(l-1)} &= c_{1,l} \cdot \ldots \cdot c_{m,l}.
\end{aligned}
$$

- Equations derived from the Weil descent

PDP algebraic model

Two sets of equations

- Equations defining symmetric polynomials

$$
\begin{aligned}
d_{1,0} &= c_{1,0} + \ldots + c_{m,0} \\
d_{1,1} &= c_{1,1} + \ldots + c_{m,1} \\
\ldots \\
d_{m,m(l-1)} &= c_{1,l} \cdot \ldots \cdot c_{m,l}.
\end{aligned}
$$

- Equations derived from the Weil descent

The system is commonly solved using Gröbner basis methods.

Logical cryptanalysis

Using SAT solvers as a cryptanalytic tool requires expressing the cryptographic problem as a Boolean formula in conjunctive normal form (CNF) - a conjunction ($\wedge$) of OR-clauses.

Example.

$$(\neg x_1 \vee x_2) \wedge$$
$$(\neg x_2 \vee x_4 \vee \neg x_5)) \wedge$$
$$(x_5 \vee x_6)$$

From the algebraic model to the SAT-reasoning model

XOR-enabled SAT solvers are adapted to read a formula in CNF-XOR form - a conjunction ($\wedge$) of OR-clauses and XOR-clauses.

Example.

$$(\neg x_1 \vee x_2) \wedge$$
$$(\neg x_2 \vee x_4 \vee \neg x_5)) \wedge$$
$$(x_1 \oplus x_5 \oplus x_6)$$

From the algebraic model to the CNF-XOR model

Variables in $\mathbb{F}_2$:
$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$.

$$\mathbf{x}_1 + \mathbf{x}_2 \cdot \mathbf{x}_4 + \mathbf{x}_5 \cdot \mathbf{x}_6 + 1 = 0$$
$$\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_4 + \mathbf{x}_5 + 1 = 0$$
$$\mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_2 \cdot \mathbf{x}_4 = 0$$
$$\mathbf{x}_2 + \mathbf{x}_5 + \mathbf{x}_2 \cdot \mathbf{x}_4 + \mathbf{x}_5 \cdot \mathbf{x}_6 + 1 = 0$$
$$\mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_6 + 1 = 0$$

Propositional variables:
$x_1, x_2, x_3, x_4, x_5, x_6$ with truth values in $\{\text{TRUE}, \text{FALSE}\}$

$$(x_1 \oplus (x_2 \wedge x_4) \oplus (x_5 \wedge x_6)) \wedge$$
$$(x_1 \oplus x_2 \oplus x_4 \oplus x_5) \wedge$$
$$(x_3 \oplus x_4 \oplus (x_2 \wedge x_4) \oplus \top) \wedge$$
$$(x_2 \oplus x_5 \oplus (x_2 \wedge x_4) \oplus (x_5 \wedge x_6)) \wedge$$
$$(x_3 \oplus x_4 \oplus x_6)$$

Multiplication in $\mathbb{F}_2$ $(\cdot)$ becomes the logical AND operation $(\wedge)$ and addition in $\mathbb{F}_2$ $(+)$ becomes the logical XOR $(\oplus)$.

From the algebraic model to the CNF-XOR model

Add new variable $x_{2,4}$ to substitute the conjunction $x_2 \wedge x_4$.

Transform the constraint

$$x_{2,4} \Leftrightarrow (x_2 \wedge x_4)$$

into CNF.

From the algebraic model to the CNF-XOR model

Propositional variables:

x_1, x_2, x_3, x_4, x_5, x_6, $x_{2,4}$, $x_{5,6}$ with truth values in $\{\text{TRUE}, \text{FALSE}\}$

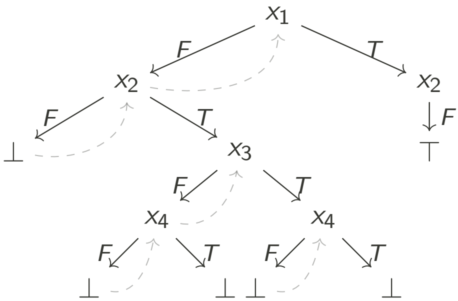
$$(x_1 \oplus (x_2 \wedge x_4) \oplus (x_5 \wedge x_6)) \wedge$$
$$(x_1 \oplus x_2 \oplus x_4 \oplus x_5) \wedge$$
$$(x_3 \oplus x_4 \oplus (x_2 \wedge x_4) \oplus \top) \wedge$$
$$(x_2 \oplus x_5 \oplus (x_2 \wedge x_4) \oplus (x_5 \wedge x_6)) \wedge$$
$$(x_3 \oplus x_4 \oplus x_6)$$

$$(\neg x_{2,4} \vee x_2) \wedge$$
$$(\neg x_{2,4} \vee x_4) \wedge$$
$$(\neg x_2 \vee \neg x_4 \vee x_{2,4}) \wedge$$
$$(\neg x_{5,6} \vee x_5) \wedge$$
$$(\neg x_{5,6} \vee x_6) \wedge$$
$$(\neg x_5 \vee \neg x_6 \vee x_{5,6}) \wedge$$
$$(x_1 \oplus x_{2,4} \oplus x_{5,6}) \wedge$$
$$(x_1 \oplus x_2 \oplus x_4 \oplus x_5) \wedge$$
$$(x_3 \oplus x_4 \oplus x_{2,4} \oplus \top) \wedge$$
$$(x_2 \oplus x_5 \oplus x_{2,4} \oplus x_{5,6}) \wedge$$
$$(x_3 \oplus x_4 \oplus x_6)$$

WDSat algorithm

Based on the Davis-Putnam-Logemann-Loveland (DPLL) algorithm.

Building a binary search-tree of height equivalent (at worst) to the number of variables.



WDSat - Three reasoning modules

CNF module

Performs unit propagation on CNF-clauses.

XORSET module

Performs unit propagation on the parity constraints. When all except one literal in a XOR clause is assigned, we infer the truth value of the last literal according to parity reasoning.

XORGAUSS module

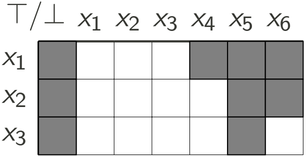
Performs Gaussian elimination on the XOR system.

WDSat - XORGAUSS module

- All variables in an XOR-clause belong to the same equivalence class.
- We choose one literal from the equivalence class to be the representative.
- Property: a representative of an equivalence class will never be present in another equivalence class.

XOR-clauses	Equivalence classes
$x_1 \oplus x_4 \oplus x_5 \oplus x_6$	$x_1 \Leftrightarrow x_4 \oplus x_5 \oplus x_6 \oplus \top$
$x_1 \oplus x_2 \oplus x_4 \oplus \top$	$x_2 \Leftrightarrow x_5 \oplus x_6 \oplus \top$
$x_2 \oplus x_3 \oplus x_6 \oplus \top$	$x_3 \Leftrightarrow x_5 \oplus \top$

- Implementation: A compact *EC* structure.

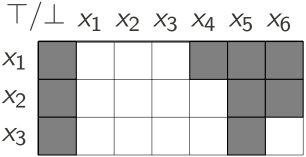


WDSat - XORGAUSS module

- All variables in an XOR-clause belong to the same equivalence class.
- We choose one literal from the equivalence class to be the representative.
- Property: a representative of an equivalence class will never be present in another equivalence class.

	XOR-clauses	Equivalence classes
	$x_1 \oplus x_4 \oplus x_5 \oplus x_6$	$x_1 \Leftrightarrow x_4 \oplus x_5 \oplus x_6 \oplus \top$
$x_2 \oplus x_5 \oplus x_6$	~~$x_1 \oplus x_2 \oplus x_4 \oplus \top$~~	$x_2 \Leftrightarrow x_5 \oplus x_6 \oplus \top$
	$x_2 \oplus x_3 \oplus x_6 \oplus \top$	$x_3 \Leftrightarrow x_5 \oplus \top$

- Implementation: A compact *EC* structure.

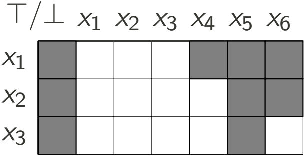


WDSat - XORGAUSS module

- All variables in an XOR-clause belong to the same equivalence class.
- We choose one literal from the equivalence class to be the representative.
- Property: a representative of an equivalence class will never be present in another equivalence class.

	XOR-clauses	Equivalence classes
	$x_1 \oplus x_4 \oplus x_5 \oplus x_6$	$x_1 \Leftrightarrow x_4 \oplus x_5 \oplus x_6 \oplus \top$
$x_2 \oplus x_5 \oplus x_6$	~~$x_1 \oplus x_2 \oplus x_4 \oplus \top$~~	$x_2 \Leftrightarrow x_5 \oplus x_6 \oplus \top$
$x_3 \oplus x_5$	~~$x_2 \oplus x_3 \oplus x_6 \oplus \top$~~	$x_3 \Leftrightarrow x_5 \oplus \top$

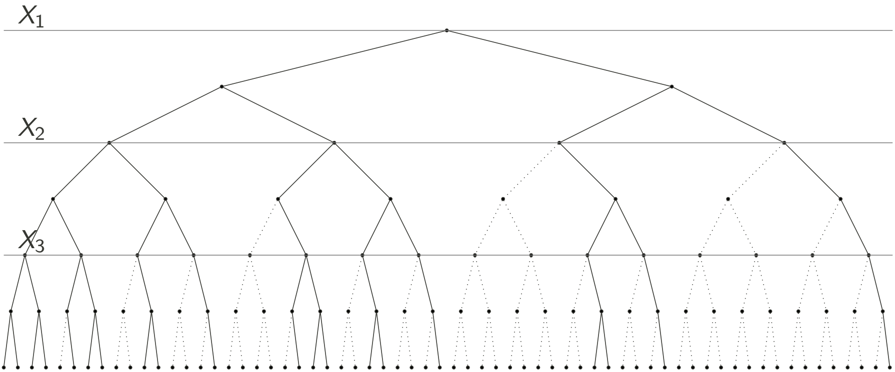
- Implementation: A compact *EC* structure.



WDSat - breaking symmetry

- Exploit the symmetry of Semaev's summation polynomials: when $X_1, ..., X_m$ is a solution, all permutations of this set are a solution as well.
- Establish the following constraint $X_1 \leq X_2 \leq \ldots \leq X_m$.
- Implement constraint in the solver using a tree-pruning-like technique.
- Optimize the complexity by a factor of $m!$.

WDSat - breaking symmetry



Experimental results

Fourth summation polynomial
Number of Boolean variables: 51, number of equations: 52.

Solving approach	SAT		UNSAT	
	Runtime (s)	#Conflicts	Runtime (s)	#Conflicts
Gröbner	229.3	*N/A*	229.4	*N/A*
MiniSat	239.7	1840190	517.0	3433304
Glucose	189.2	1527158	274.8	2056575
MapleLCMDistChronoBT	655.1	4035131	918.7	5378945
CaDiCaL	43.6	254194	141.3	629869
CryptoMiniSat	331.8	1791188	707.9	3416526
WDSat+br-sym	**0.24**	**19166**	**0.63**	**44034**

Table: Comparing Gröbner basis and SAT-based approaches for solving the point decomposition problem. Running times are in seconds.

Perspectives

- Understand the complexity of CNF-XOR instance solving.
- Combine WDSAT with CDCL techniques.
- Use WDSAT for attacks on other cryptosystems.
- Understand the link between algebraic and SAT-based solving methods.