NEURAL INFORMATION PROCESSING SYSTEMS

ReThinkLab SJTU

SHANGHAI JIAO TONG UNIVERSITY

Learning Plaintext-Ciphertext Cryptographic Problems via ANF-based SAT Instance Representation

Advances in Neural Information Processing Systems (NeurIPS 2024)

Xinhao Zheng, Yang Li, Cunxin Fan, Huaijin Wu,
Xinhao Song, Junchi Yan

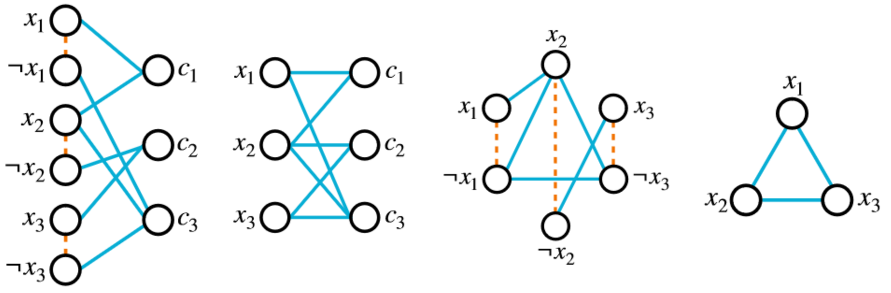
Shanghai Jiao Tong University

SAT for Cryptanalysis

Round function

$$T_1 = (L_i \lll a) \& (L_i \lll b),\ T_2 = L_i \lll c$$
$$T_3 = T_1 \oplus T_2 \oplus K_i$$
$$L_{i+1} = T_3 \oplus R_i$$
$$R_{i+1} = L_i$$

Cryptographic Problem
(like Plaintext-Ciphertext Problem)



(a) LCG (b) VCG (c) LIG (d) VIG

$$(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$

Satisfiability Problem
in CNF formula

ANF: ∧, ⊕

l-length Clause in ANF

CNF: ¬, ∧, ∨

Nearly 2^l-length Clause in CNF

$$x' \vee \neg x_2 \vee \neg x_3$$
$$\neg x' \vee x_2 \qquad \Longleftrightarrow \qquad x' + x_2 x_3 = 0$$
$$\neg x' \vee x_3$$

Problem size ballooning considerably

increasing computational complexity,
losing high-order operational information.

ANF formula for Crypto Operations

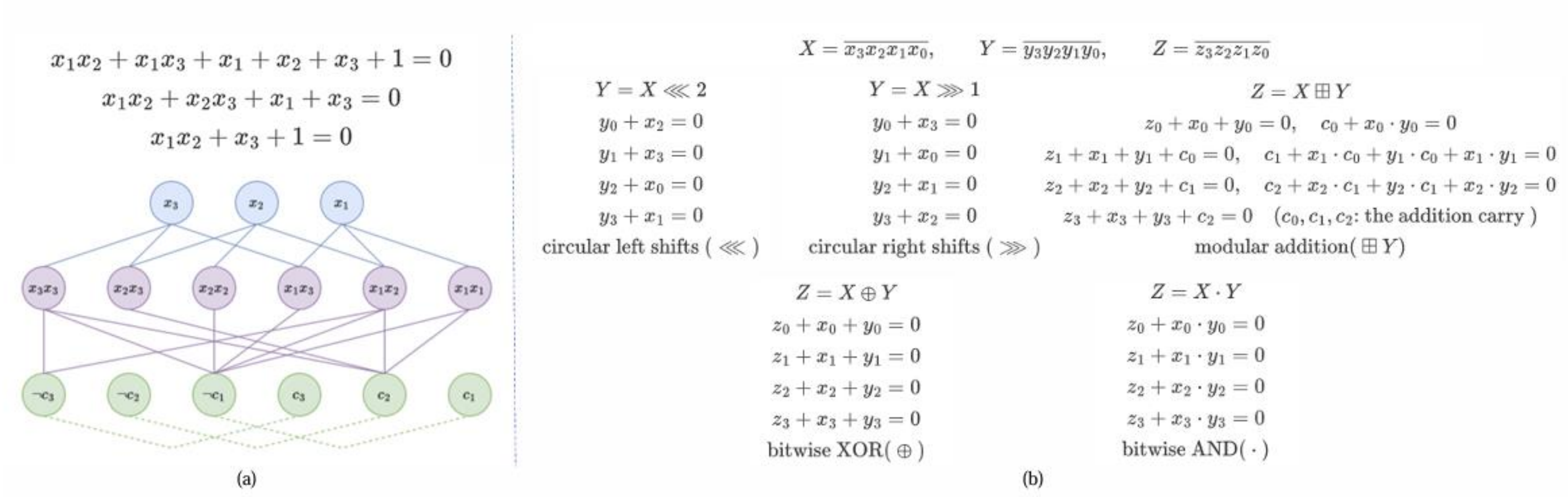


Figure 1: (a) Example ANF formula graph from MQ problem; (b) The transformations to express the circular left shifts ($\lll$), circular right shifts ($\ggg$), modular addition ($\boxplus$), bitwise XOR ($\oplus$), and bitwise AND ($\cdot$) operations in ANF.

ANF-based Graph Structures

A single iteration consists of the following two updates

Literal to clause

$$
\begin{aligned}
L_{l2c}^{(t)} &= L_{l2l}([L^{(t)}[I], L^{(t)}[J]]) \\
[C_{m,\text{pos}}^{(t)}, C_{m,\text{neg}}^{(t)}] &= M_{l2c}^{T} L_{\text{msg}}(L_{l2c}^{(t)}) \\
(C_{\text{pos}}^{(t+1)}, C_{h,\text{pos}}^{(t+1)}) &\leftarrow C_{u,\text{pos}}([C_{h,\text{pos}}^{(t)}, C_{\text{neg}}^{(t)}, C_{m,\text{pos}}^{(t)}]) \\
(C_{\text{pos}}^{(t+1)}, C_{h,\text{neg}}^{(t+1)}) &\leftarrow C_{u,\text{neg}}([C_{h,\text{neg}}^{(t)}, C_{\text{pos}}^{(t)}, C_{m,\text{neg}}^{(t)}])
\end{aligned}
$$

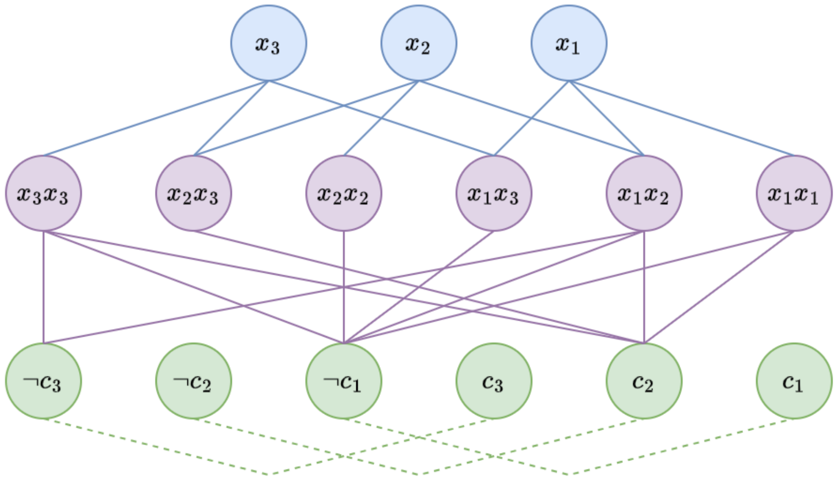
Clause to literal

$$
\begin{aligned}
L_{c2l}^{(t)} &= M_{l2c} C_{\text{msg}}([C_{\text{pos}}^{(t)}, C_{\text{neg}}^{(t)}]) \\
L_{m}^{(t)} &= M_{l2l} L_{l2m}(L_{c2l}^{(t)}) \\
(L^{(t+1)}, L_{h}^{(t+1)}) &\leftarrow L_{u}([L_{h}^{(t)}, L_{m}^{(t)}])
\end{aligned}
$$

$$x_1x_2 + x_1x_3 + x_1 + x_2 + x_3 + 1 = 0$$

$$x_1x_2 + x_2x_3 + x_1 + x_3 = 0$$

$$x_1x_2 + x_3 + 1 = 0$$



We only save the embedding of vanilla literals and pairs of complementary clauses.

Complexity for CNF and ANF formula

Table 1: Parameters of SAT problems in CNF and ANF

	Datasets	SR(5)	SR(25)	Scipher 3-8-16	Scipher 3-16-32	Scipher 6-8-16	Scipher 6-16-32	Speck 3-8-16	Speck 6-8-16
CNF	#Literals	6	424	25	49	49	97	57	129
	#Clauses	75	5492	195	225	735	1519	336	921
	#Nodes	87	6340	245	323	833	1713	450	1179
ANF	#Literals	5	25	24	48	48	96	56	128
	#Clauses	11	26	24	48	48	96	64	136
	#Nodes	27	77	72	144	144	288	184	400

Key solving algorithm for Plaintext-Ciphertext Problem

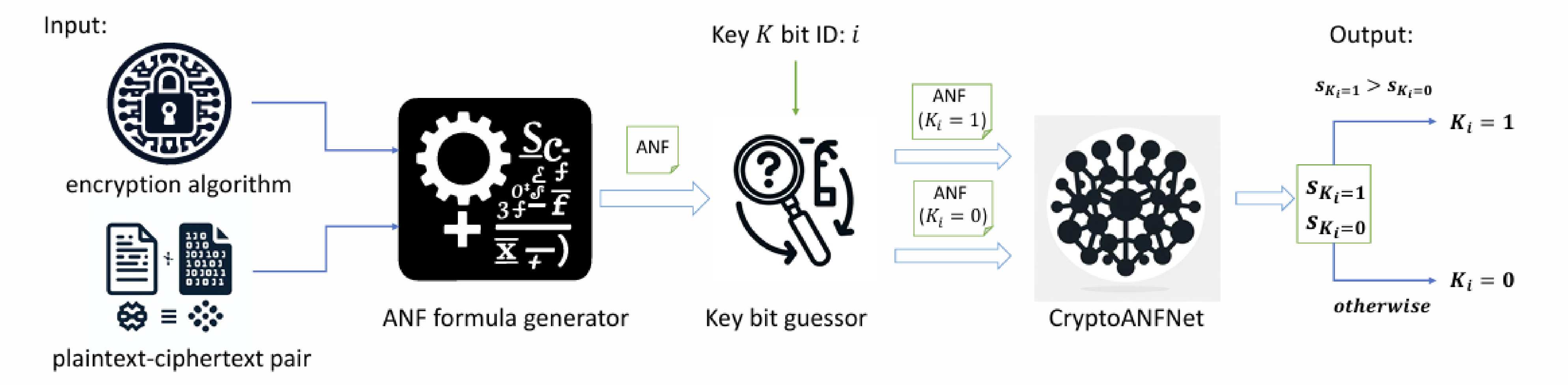


Figure 2: **The pipeline of the key-solving algorithm.** Given a plaintext-ciphertext pair and an encryption algorithm, we first transform them into an ANF-based instance of an MQ problem. Then, for a specific key bit K_i (the i-th bit of key), we guess its value as either 0 or 1 and generate two derived SAT instances. We then employ CryptoANFNet to predict the satisfiability of each instance. The final determination of K_i is based on which instance receives a higher satisfiability score.

Experimental Results for Satisfiability Prediction

Table 2: Performance of different learning-based solvers on synthetic datasets

Datasets	SR(5)	SR(25)	Scipher 3-8-16	Scipher 3-16-32	Scipher 6-8-16	Scipher 6-16-32	Speck 3-8-16	Speck 6-8-16
NeuroSAT	91.0%	57.0%	74.0%	72.7%	53.0%	51.0%	55.0%	52.5%
CryptoANFNet	**96.0%**	**72.0%**	**76.5%**	**75.6%**	**69.0%**	**66.5%**	**72.0%**	**68.5%**

Table 3: Performance for key-solving algorithm in solving MQ problems on synthetic datasets.

Datasets	Scipher 3-8-16	Scipher 3-16-32	Scipher 6-8-16	Scipher 6-16-32	Speck 3-8-16	Speck 6-8-16
NeuroSAT	74.0%	72.7%	53.0%	51.0%	55.0%	52.5%
CryptoANFNet	76.5%	75.6%	69.0%	66.5%	72.0%	68.5%
CryptoANFNet+ key-solving	**82.0%**	**78.4%**	**70.0%**	**69.0%**	**75.0%**	**71.0%**

Experimental Results for Efficiency Comparison

Table 4: Comparing the efficiency of different solvers for solving the MQ problem on synthetic datasets. (Average runtime: (SAT, UNSAT) ms/instance)

Datasets	SR(5)	SR(25)	Scipher 3-8-16	Scipher 3-16-32	Scipher 6-8-16	Scipher 6-16-32	Speck 3-8-16	Speck 6-8-16
NeuroSAT [5]	(3,3)	(20,20)	(7,7)	(10,10)	(7,7)	(14,14)	(13,13)	(18,18)
CryptoANFNet	(2,2)	(5,5)	(8,8)	(9,9)	(10,10)	(8,8)	(11,11)	(14,14)
WDSat [42]	(36,34)	(2470,5662)	(38,38)	(39,39)	(40,37)	(86,150)	(44,47)	(46,46)
CryptoMiniSat [44]	(4,4)	(13491,35912)	(4,4)	(7,9)	(8,9)	(410,1354)	(5,5)	(6,8)
Kissat [45]	(2,2)	(4922,14856)	(2,2)	(2,2)	(5,8)	(219,464)	(3,3)	(4,5)

Table 5: Comparing the efficiency of incomplete solvers for solving the MQ problem on synthetic datasets. (Average runtime: (SAT, UNSAT) ms/instance)

Datasets	SR(5)	SR(25)	Scipher 3-8-16	Scipher 3-16-32	Scipher 6-8-16	Scipher 6-16-32	Speck 3-8-16	Speck 6-8-16
WalkSAT [46]	(3,640)	(762,744)	(4,6)	(10,12)	(289,26)	(831,899)	(39,480)	(482,538)
RoundingSAT [47]	(3,3)	(36758,50122)	(3,5)	(7,10)	(28,20)	(664,1801)	(23,24)	(29,35)
FourierSAT [48]	(1275,8670)	(9620,9687)	(983,426)	(1779,459)	(8163,416)	(8830,8862)	(8733,8689)	(8799,8912)

NEURAL INFORMATION PROCESSING SYSTEMS

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Thank you!

- Contact:
  - Xinhao Zheng :
    void_zxh@sjtu.edu.cn
  - Prof. Junchi Yan
    yanjunchi@sjtu.edu.cn

- Homepage of our lab:
  http://thinklab.sjtu.edu.cn/