Local Inversion of maps: Application to Cryptanalysis

Virendra Sule[1]

[1]Department of Electrical Engineering
Indian Institute of Technology Bombay, India
(vrs@ee.iitb.ac.in)

Industrial Computeralgebra Conference with focus Cryptography, September 28, 2021
Carl von Ossietzky University, Oldenburg, Germany

Cryptanalysis of Cipher Algorithms

- Estimating computational resources (time and memory) to determine unknown information about the cipher algorithm (e.g. symmetric key or internal states) from known information (ciphertexts or outputs streams).
- Modern ciphers have symmetric key lengths > 80 bits. Algebraic models tend to be highly complex due to latent and internal variables.
- It is necessary to find all possible solutions of equations (unknown keys) for given (limited) data of ciphertext for known and chosen plaintext.
- Finding all solutions of nonlinear equations in finite fields efficiently, is an "Unfinished Agenda" in Computational Sciences. Need for representation of all solutions.

Known and Unknown in cipher algorithms

- Block cipher: Encryption function $C = E(K, P)$. Known: (P, C) blocks. Unknown: K symmetric key.
- Stream cipher: Dynamical system with output (F, f). F state update map $F : X \mapsto X$, X state space. f output map $f : X \mapsto \mathbb{F}_2$. Known output stream

$$\hat{w} = [f(x), f(F(x)), f(F^{(2)}(x), \ldots, f(F^{(n-1)}(x))]^T$$

  Unknown internal state x.
- Stream cipher key recovery: Initial state $x(0) = (K, IV)$, K symmetric key unknown, IV known. From internal state $x(k_0)$ to recover $x(0)$.
- Cryptanalysis algorithm is required to find all solutions to the equations of constraints. However number of such solutions is likely to be very small if sufficient data is available.

Local Inversion problem

- A general form which captures most such problems is the

Local Inversion Problem

Given a map $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ and y in $\mathbb{F}_2^n$. Find all x in $\mathbb{F}_2^n$ such that $F(x) = y$. (In char 2).

- Global Inversion: 1) To find whether $F : \mathbb{F}_q^n \mapsto \mathbb{F}_q^n$ is a permutation, 2) When F is a permutation, find the inverse map $G : \mathbb{F}_q^n \mapsto \mathbb{F}_q^n$ such that $F \circ G = G \circ F = Id_{\mathbb{F}_q^n}$.

Remark

1),2) Solved by linear representation (LR) of F. LR of G in the same basis as F when it is a permutation is shown in the paper [1], `arxiv.org/cs.SY/2010.14601`. When F is not a permutation, G does not exist hence LR does not solve the Local Inversion problem.

Time Memory Tradeoff (TMTO) Attack

- Classic algorithm for inversion of maps, $F : S \mapsto S$ where S is a finite set.
- Probabilistic with success probability of Birthday attack 63% for number of time steps and storage of $\sqrt{(|S|)}$ size. Not scalable for realistic sizes of inversion when $|S|$ is exponential. Known under the name Rainbow table attack.
- TMTO does not use any structure of vector spece on S over finite field and hence on map F. If such a structure is used we get the proposed algorithms for local inversion which are far more scalable and efficient.

A dynamical system induced by F

Local inversion requires knowledge of dynamics of iterations.

- Map F induces the dynamical system in $\mathbb{F}_2^n$

$$x(k+1) = F(x(k)), k = 0, 1, 2, \ldots$$

- Trajectories of the system:
  1. Closed (periodic) orbits: x is in a periodic orbit of period N if $x = F^{(N)}(x)$. The periodic orbit is the sequence, $S(F, x) = \{x, F(x), \ldots, F^{(N-1)}(x)\}$
  2. Chains of length l, $\{x, F(x), \ldots, F(l)(x)\}$ of whose only the terminal point $F^{(l)}(x)$ is in a periodic orbit.
  3. Fixed points $x = F(x)$ periodic orbits of period 1.
  4. Garden of Eden of F, $GOE = \{x \notin \text{im } F\}$.

Existence and multiplicity of solutions

Proposition

Equation $y = F(x)$ has no solution iff y belongs to GOE

Theorem: All possible solutions

- $F(x) = y$ has solution in a periodic orbit P iff y belongs to P. Such a periodic orbit is unique.
- All other possible solutions belong to the chains $F^k(z), k \geq 1$ for z in the GOE of F.
- If y is neither in a periodic orbit nor in the GOE, then solutions arise in some of the segments $F^{(k)}(z), 1 < k \leq (l-1)$ for z in the GOE.
- $y = F(x)$ has a unique solution iff y belongs exclusively to a chain segment of a unique point in GOE or to a unique periodic orbit

Computation of solutions

- Searching for the unique periodic orbit of period N which contains y. Solving $x = F^{(N-1)}(y)$.
- Searching for all chains $F^{(k)}(z)$, z in GOE which contain y.
- Dont search if y belongs to GOE.

Complete Algorithm: Offline and Online computation

- *Offline computation*: which depends on F but does not involve y.
  1. Set Π of all possible periods of closed orbits of F. (NP Hard).
  2. GOE of F. (NP Hard).
- *Online computation*: Involves y and information on F gathered in offline computation.
  1. Search for N in Π such that $F^{(N)}(y) = y$. (Polynomial time and parallel for each N).
  2. Search for z in GOE such that $F^{(k)}(z) = y$ while not already in a periodic orbit. (Polynomial time and parallel for each z).

The Big Picture: Solution using Linear Complexity (LC) I

- Checking the periodicy condition $y = F^{(N)}(y)$ to find the inverse x is not necessary and can be replaced by discovering the LC of the sequence

$$S(F, y) = \{y, F(y), F^{(2)}(y), \dots\}$$

- LC is the smallest m such that $F^{(m+j)}(y)$ is linearly dependent on

$$\{F^{(j)}(y), F^{(1+j)}(y), \dots, F^{(m-1+j)}(y)\}$$

The Big Picture: Solution using Linear Complexity (LC) II

- (LC) was proposed in literature for representing a sequence by a smallest degree LFSR generated sequence.

New Idea

The big idea being proposed here is to show that if the sequence is generated by iterates of a map F then computation of LC solves the local inversion problem.

- **Minimal polynomial:** Operator polynomial $\phi(X)$ in $\mathbb{F}_2[X]$: $X(x) = F(x)$, $(X^m + X^n)(x) = F^{(m)}(x) + F^{(n)}(x)$. $\phi(X)$ is called *annihilating* if $\phi(X)(x) = 0$. $\phi(X)$ is called *minimal* if monic and of least deg annihilating.

Solution in a periodic orbit

- **Proposition**: The minimal polynomial $m(X)$ divides any annihilating polynomial and satisfies $m(0) \neq 0$. $S(F, y)$ is periodic iff the minimal polynomial satisfies $m(0) \neq 0$ and the period is $N =$ order $m(X)$.
  For proof see the paper [3], A complete algorithm for local inversion of maps: Application to Cryptanalysis, arxiv.org/2105.07332.
- If

$$m(X) = X^m + \alpha_{(m-1)} X^{(m-1} + \ldots + \alpha_1 X + \alpha_0$$

The solution $x = F^{(N-1)}(y)$ is expressible by the formula

$$x = (1/\alpha_0)[F^{(m-1)}(y) - (\sum_{j=1}^{(m-1)} \alpha_j F^{(j-1)}(y))] \qquad (1)$$

(polynomial time computation when m is $O((\log N)^r)$).

Incomplete algorithm for small linear complexity I

- Fix a bound M of polynomial size $O(n^r)$. Compute the sequence $S(F, y) = \{F^{(k)}(y)\}$ for $k = 0, \ldots, M$.
- Compute a possible minimal polynomial $m(X)$ of degree $m \leq \lfloor M/2 \rfloor$ by locating least no. of LI vectors in $S(F, y)$ s.t. $F^{(m+j)}(y)$ is dependent on previous m vectors.
- The m in above computation is located by a Hankel matrix $H(m)$ of size m over $\mathbb{F}_2$ at which

  $$m = \operatorname{rank} H(m) = \operatorname{rank} H(m+1)$$

  The unique solution of the polynomial $m(X)$ is obtained by solving a linear system defined by $H(m)$ and the last column of $H(m+1)$.

Incomplete algorithm for small linear complexity II

- Find one solution x by the formula (1). If $F(x) = y$, the solution is verified. If x fails to be a solution increment m and repeat computation of $m(X)$ until $m = \lfloor M/2 \rfloor$.

Theorem

If minimal polynomial exists of degree $m \leq \lfloor M/2 \rfloor$ one solution x can be computed in polynomial time

Solution in a chain

- Algorithm requires the set GOE as input.
- GOE of F can be computed by solving implicants of the Boolean system in x, y such that $F(x) = y$ is NOT satisfied. Can be achieved by the Boolean solver.
- For z in GOE compute the chain $z(k) = F^{(k)}(z)$. While $z(k)$ is not in one of the periodic orbit compute $z(k+1)$ until $z(l) = y$. Then $x = F^{(l-1)}(y)$ is a solution in the chain starting from z. (Polynomial time computation for each k. Polynomial time in l the length of the chain).
- The algorithm repeated parallelly for each z.

Complexity of complete algorithm

- Offline computation: NP hard. Sets Π and GOE. Π computed by Linear Representation of the map F, [1]. GOE computed by the Boolean system solver [2, 3].
- Online computation: Polynomial order in linear complexity m (degree of minimal polynomial) of the periodic orbit and chain length l which contain y. Hence polynomial time if m and l are not exponential.
- When m is $O(n^k)$ polynomial order, the inversion algorithm can find one solution in polynomial time. (This is a disruptive breaking of the map F computable by the incomplete algorithm).
- Computation of linear complexity of $S(F, y)$ is not necessary if Π the set of all possible periods is pre-computed offline. Checking whether $y = F^{(N)}(y)$ is polynomial time for a given N.

Cryptanalysis: Block cipher

- Encryption algorithm with known plaintext P and the ciphertext C gives equation
$$C = E(K, P)$$
Then $y = C$, $F(x) = E(x, P)$.
- Decryption algorithm with chosen ciphertext C and decrypted message P gives equation
$$P = D(K, C)$$
Then $y = P$, $F(x) = D(x, C)$.
- If algorithms E or D are known, then the complete algorithm for local inversion computes all x which give all possible keys for the data (P, C).
- Incomplete algorithm can be used to find a-priori probability of breaking F for a random data.

Cryptanalysis: Stream cipher

- Stream cipher (F, f).
- Two problems:
  1. Internal state recovery from output sequence.
  2. Key recovery from internal state.
- Map for internal state recovery: Output sequence $\hat{w} = [w_1, w_2, \ldots, w_n]^T$. Internal state $x = x(k_0)$

$$\hat{w} = [f(x), f(F(x)), \ldots, f(F^{(n-1)}(x)]^T$$

  Find x given $\hat{w}$.
- Map for initial condition recovery:

$$x(0) = F^{(k_0)}(x)$$

  Key recovered from $x(0)$ when IV part of $x(0)$ matches with known IV.

Conclusions I

- Local inversion or solving $y = F(x)$ can be carried out by forward evaluation of F instead of solving algebraic equations or Boolean symbolic models of F. Makes cryptanalysis enormously scalable for realistic cases.
- Multiple solutions of $y = F(x)$ depend on the number of dynamic trajectories of iteration of F in which y belongs. There is always a unique solution in a periodic orbit containing y. All other possible solutions belong to chains.
- Theoretical advance: Solution of the inversion problem using LC. Previously application of LC was only for modeling by LFSR sequences.

Conclusions II

- When the sequence $S(F, y)$ has polynomial size linear complexity and the minimal polynomial with $m(0) \neq 0$, one solution of the inverse can be computed in polynomial time. This has most disruptive consequence in practical Cryptanalysis.
- Some estimates of polynomial sizes of LC:
  1. $AES128$, $LC \approx 128^3 = 2,097,152$.
  2. $RSA1024$ inversion without factoring $LC \approx 1024^3 = 1,073,141,824$.

  sizes of Hankel matrices of linear systems to be inverted.

References

[1] Ramachandran A. and Virendra Sule. On computation of the inverse of a polynomial map over finite fields using the reduced koopman dual linear map. *arxiv.org/cs.SY/2010.14601*, 2020.

[2] Virendra Sule. An implicant based, parallel, all solution solver for boolean satisfiability. *arxiv.org/cs.DS/1611.09590v3*, 2017.

[3] Virendra Sule. A complete algorithm for local inversion of maps: Application to cryptanalysis. *arxiv.org/cs.CR/2105.07332*, 2021.

Thank You

Thank You