2.13 %

Algebraic Attacks

Topics in algebraic attacks on cryptosystems

Hossein Hadipour
hsn.hadipour@gmail.com

University of Tehran
College of Science
School of Mathematic, Statistics and Computer Science

October 14, 2016

Outline

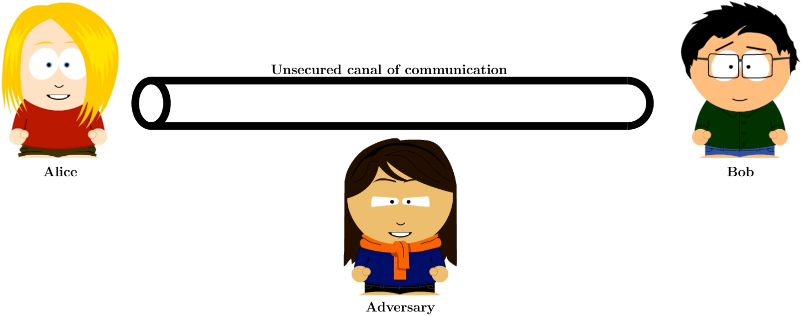
6.38 %

Outline

1. Introduction
   - Security concerns
   - Goals
2. From Cryptography to Algebra
3. Extraction of equations
   - Algebrization of CTC
   - Algebrization of stream ciphers
   - Algebrization of public key cryptosystems
4. Solvers
   - linearization and relinearization method
   - Solving equations with Groebner and Border basis
   - Mixed Integer Linear Programming (MILP)
   - SAT Solver

Security concerns

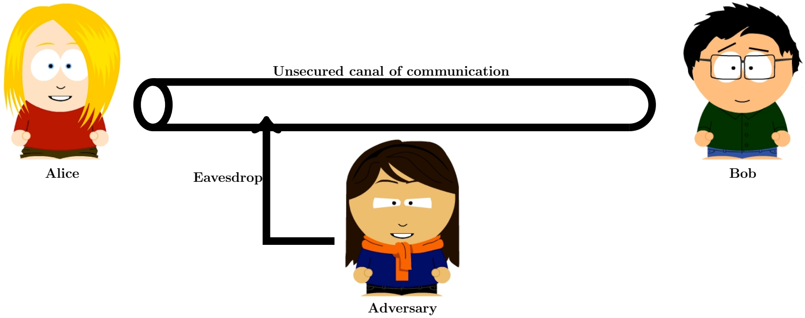
Historically, cryptography arose as a means to enable parties to maintain privacy of the information they send to each other, even in the presence of an adversary with access to the communication channel.



Security concerns

Main problems:

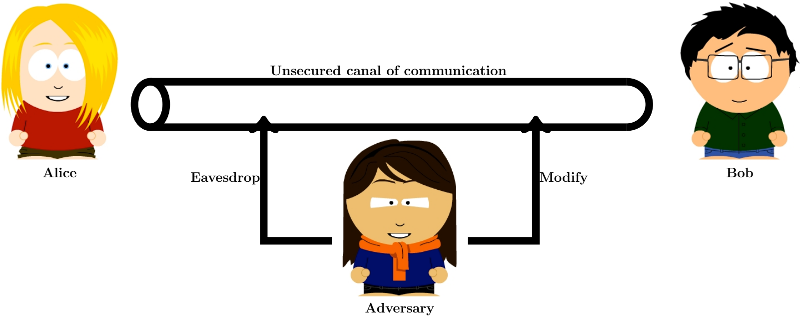
- Eavesdropping



Security concerns

Main problems:

- Eavesdropping
- Modifying

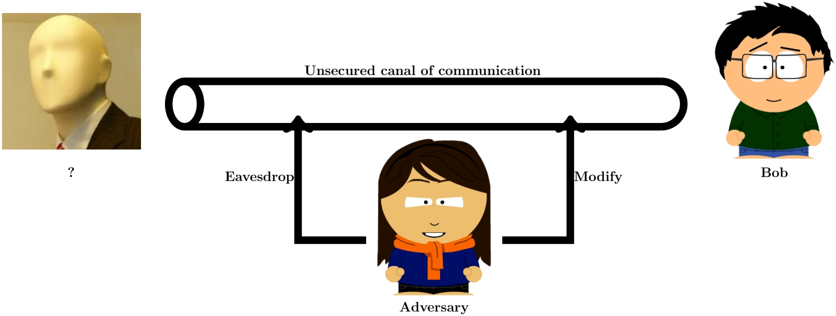


14.89 %

Security concerns

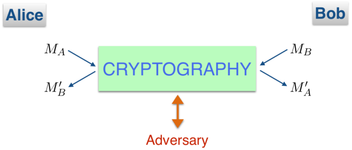
Main problems:

- Eavesdropping
- Modifying
- Impersonation



The main goals of cryptography

Cryptography is used to build a secure channel



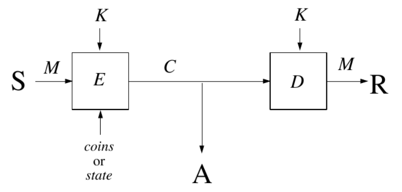
Main Goals

- **privacy or secrecy**: Adversary does not learn anything about M_A, M_B
- **message integrity (or message authentication)** : each party should be able to identify when a message it receives was sent by the party claiming to send it, and was not modified in transit.

The symmetric setting

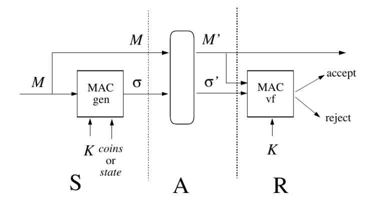
Symmetric encryption

providing privacy



Message Authentication Code

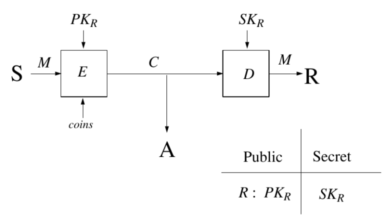
providing authentication



The asymmetric setting

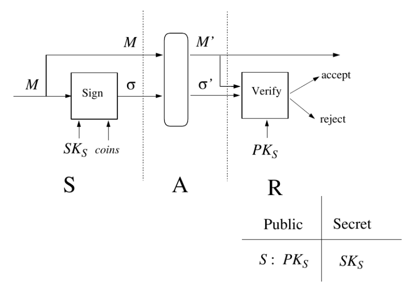
Asymmetric encryption

providing privacy



Digital signature

providing authentication



23.4 %

Outline

1. Introduction
   - Security concerns
   - Goals
2. From Cryptography to Algebra
3. Extraction of equations
   - Algebrization of CTC
   - Algebrization of stream ciphers
   - Algebrization of public key cryptosystems
4. Solvers
   - linearization and relinearization method
   - Solving equations with Groebner and Border basis
   - Mixed Integer Linear Programming (MILP)
   - SAT Solver

Definition (cryptosystem)

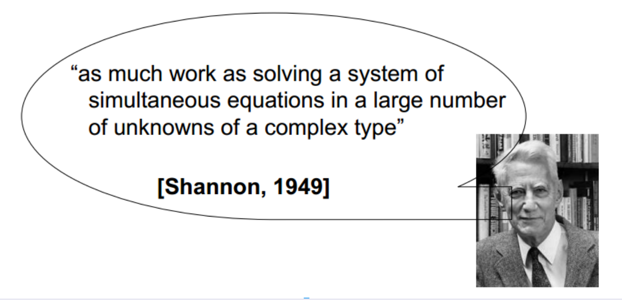
A cryptosystem (or cipher or encryption scheme) consists of the following parts:

1. A set $\mathcal{P}$ called the set of *plaintext units*.
2. A set $\mathcal{C}$ called the set of *ciphertext units*.
3. A set $\mathcal{K}$ called the *key space*.
4. For every key $k \in \mathcal{K}$, an *encryption map* $E_k : \mathcal{P} \to \mathcal{C}$.
5. For every key $k \in \mathcal{K}$, a *decryption map* $D_k : \mathcal{C} \to \mathcal{P}$.
6. A map $\eta : \mathcal{K} \to \mathcal{K}$ such that $D_{\eta(k)} \circ E_k = id_{\mathcal{K}}$ for all $k \in \mathcal{K}$. A pair $(k, \eta(k))$ is called a *key pair*.

The cryptosystem is called symmetric if $\eta(k)$ can be computed efficiently given the knowledge of k and E_k. Otherwise, the system is called a public key cryptosystem.

Shanon's idea about algebraic attacks

Breaking a good cipher should require:



29.79 %

The basis of all algebraic attacks

The following theorem is the basis of all algebraic attacks.

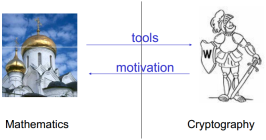
Theorem

Over a finite field K, every map $\phi : K^n \to K^m$ is polynomial, i.e. there exist polynomials $f_1, \ldots, f_m \in K[x_1, \ldots, x_n]$ such that

$$\phi(a_1, \ldots, a_n) = (f_1(a_1, \ldots, a_n), \ldots, f_m(a_1, \ldots, a_n)),$$

for all $a_1, \ldots, a_n \in K$.

since any encryption map between finite dimentional vector spaces over finite field is polynomial, it is natural to represent the task of breaking a cryptosystem by the problem of solving a multivariate polynomial system of equation over finite field.



What are Algebraic Attacks?

algebraic attacks consist of 3 step:

What are Algebraic Attacks?

algebraic attacks consist of 3 step:

1. expressing the cipher operations as a system of equations

What are Algebraic Attacks?

algebraic attacks consist of 3 step:

1. expressing the cipher operations as a system of equations
2. substituting in known data for some of the variables

What are Algebraic Attacks?

algebraic attacks consist of 3 step:

1. expressing the cipher operations as a system of equations
2. substituting in known data for some of the variables
3. solving for the key or plaintext

What are Algebraic Attacks?

algebraic attacks consist of 3 step:

1. expressing the cipher operations as a system of equations
2. substituting in known data for some of the variables
3. solving for the key or plaintext

We can break any cipher, by this attack if we can solve equation.

34.04 %

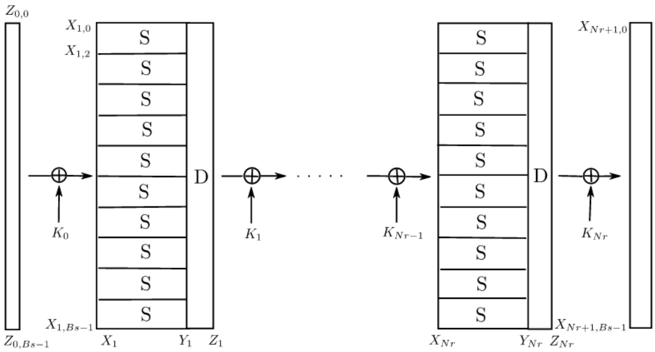
Outline

1. Introduction
   - Security concerns
   - Goals
2. From Cryptography to Algebra
3. Extraction of equations
   - Algebrization of CTC
   - Algebrization of stream ciphers
   - Algebrization of public key cryptosystems
4. Solvers
   - linearization and relinearization method
   - Solving equations with Groebner and Border basis
   - Mixed Integer Linear Programming (MILP)
   - SAT Solver

Structure of CTC

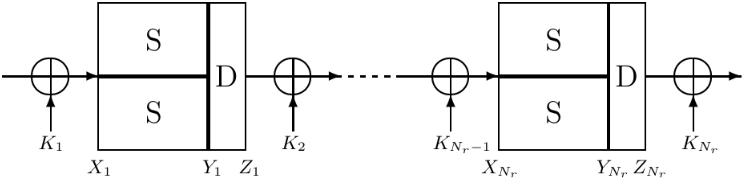
The Courtois Toy Cipher

The CTC has been designed by Nicolas Courtois to apply and test methods of algebraic cryptanalysis.



Parameters of CTC

- The number of rounds is N_r.
- Let $B = 1, 2, ..., 128$ be the number of S-boxes in each round. There are $B \cdot s$ bits in each round
- The key size is equal to the block size.
- $K_0 = (K_{01}, ..., K_{0 \cdot Bs})$ is the secret key.
- We denote X_{ij} , for $i = 1, ..., N_r$, $j = 0, ..., B \cdot s - 1$, the inputs of the i-th round after the XOR with the derived key.
- We denote Z_{ij} , for $i = 1, ..., N_r$, $j = 0, ..., B \cdot s - 1$, the outputs of the i-th round before the XOR with the next derived key.
- $Z_0 = plaintext$, $X_{N_r+1} = ciphertext$



Key Schedule and the Diffusion Layer of CTC

- There is no S-boxes in the key schedule and the derived key in round i, K_i is obtained from the secret key K_0 , by a very simple permutation of wires:

$$K_{ij} := K_{0(i+j \mod B \cdot s)}$$

- The diffusion part D of the cipher is defined as follows:

$$\begin{cases} Z_{i(257 \mod Bs)} = Y_{i0} & \forall\ i = 1, ..., N_r \\ Z_{i(j \cdot 1987+257 \mod Bs)} = Y_{ij} \oplus Y_{i(j+137 \mod Bs)} & \forall\ j \neq 0 \forall\ i. \end{cases}$$

In general the diffusion layer gives rise to linear equations.

S-Box I

The S-box is a non-linear operation. However, finding equations is still easy.

As an example consider the 3-bit (since it fits on the slides) S-box

$$[7, 6, 0, 4, 2, 5, 1, 3].$$

Construct the matrix on the right and perform fraction-free Gaussian elimination on it (fitting a linear model).

$$
\begin{array}{c}
\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & x_0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & x_1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & x_2 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & y_0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & y_1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & y_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & x_0x_1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & x_0x_2 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & x_0y_0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & x_0y_1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & x_0y_2 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & x_1x_2 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & x_1y_0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_1y_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & x_1y_2 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & x_2y_0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & x_2y_1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & x_2y_2 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & y_0y_1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & y_0y_2 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & y_1y_2
\end{pmatrix}
\end{array}
$$

S-Box II

$$
\left(\begin{array}{ccccccccr}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_0y_0 + x_1 + x_2 + y_0 + y_1 + 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x_0y_0 + x_0 + x_1 + y_2 + 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & x_0y_0 + x_0 + y_0 + 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & x_0y_0 + x_0 + x_2 + y_1 + y_2 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & x_0y_0 + x_0 + x_1 + x_2 + y_0 + y_1 + y_2 + 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & x_0y_0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & x_0y_0 + x_2 + y_0 + y_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_0y_0 + x_1 + y_1 + 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0x_2 + x_1 + y_1 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0x_1 + x_1 + x_2 + y_0 + y_1 + y_2 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0y_1 + x_0 + x_2 + y_0 + y_2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0y_0 + x_0y_2 + x_1 + x_2 + y_0 + y_1 + y_2 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_1x_2 + x_0 + x_1 + x_2 + y_2 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0y_0 + x_1y_0 + x_0 + x_2 + y_1 + y_2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0y_0 + x_1y_1 + x_1 + y_1 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_1y_2 + x_1 + x_2 + y_0 + y_1 + y_2 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_0y_0 + x_2y_0 + x_1 + x_2 + y_1 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_2y_1 + x_0 + y_1 + y_2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x_2y_2 + x_1 + y_1 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{y_0y_1 + x_0 + x_2 + y_0 + y_1 + y_2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{y_0y_2 + x_1 + x_2 + y_0 + y_1 + 1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{y_1y_2 + x_2 + y_0}
\end{array}\right)
$$

S-Box III

Computing S-box equations by sagemath (`http://www.sagemath.org`):

```
sage: S = mq.SBox(7,6,0,4,2,5,1,3)
sage: S.polynomials()
[x0*x2 + x1 + y1 + 1,
 x0*x1 + x1 + x2 + y0 + y1 + y2 + 1,
 x0*y1 + x0 + x2 + y0 + y2,
 x0*y0 + x0*y2 + x1 + x2 + y0 + y1 + y2 + 1,
 x1*x2 + x0 + x1 + x2 + y2 + 1,
 x0*y0 + x1*y0 + x0 + x2 + y1 + y2,
 x0*y0 + x1*y1 + x1 + y1 + 1,
 x1*y2 + x1 + x2 + y0 + y1 + y2 + 1,
 x0*y0 + x2*y0 + x1 + x2 + y1 + 1,
 x2*y1 + x0 + y1 + y2,
 x2*y2 + x1 + y1 + 1,
 y0*y1 + x0 + x2 + y0 + y1 + y2,
 y0*y2 + x1 + x2 + y0 + y1 + 1,
 y1*y2 + x2 + y0]
```

Breaking CTC(B = 1, Nr = 1) I

Example

For an illustration how to put these equations together consider the following example for $B = 1$ and $N_r = 1$.

The initial key addition is expressed through:

$$\begin{aligned} 0 =& K_{00} + Z_{00} + X_{00}, \\ 0 =& K_{01} + Z_{01} + X_{01}, \\ 0 =& K_{02} + Z_{02} + X_{02}, \end{aligned}$$

The key addition of the first round:

$$\begin{aligned} 0 =& K_{10} + Z_{10} + X_{20}, \\ 0 =& K_{11} + Z_{11} + X_{21}, \\ 0 =& K_{12} + Z_{12} + X_{22}. \end{aligned}$$

The diffusion layer consists of three linear equations:

$$\begin{aligned} 0 =& Z_{10} + Y_{11} + Y_{10}, \\ 0 =& Z_{11} + Y_{12} + Y_{11}, \\ 0 =& Z_{12} + Y_{10}. \end{aligned}$$

The key schedule equations:

$$\begin{aligned} 0 =& K_{10} + K_{01}, \\ 0 =& K_{11} + K_{02}, \\ 0 =& K_{12} + K_{00}. \end{aligned}$$

Breaking CTC(B = 1, Nr = 1) II

Example

The S-box is represented as:

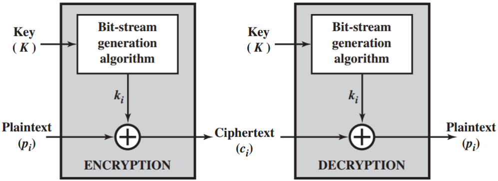
$$
\begin{aligned}
0 =&1 + Y_{10} + X_{12} + X_{11} + X_{10} + X_{10}X_{11},\\
0 =&1 + Y_{11} + X_{11} + X_{10}X_{12},\\
0 =&1 + Y_{11} + X_{11} + X_{10}Y_{10},\\
0 =&Y_{11} + Y_{10} + X_{12} + X_{10}Y_{11},\\
0 =&1 + Y_{12} + Y_{11} + Y_{10} + X_{11} + X_{11}X_{12} + X_{10},\\
0 =&1 + Y_{12} + Y_{11} + Y_{10} + X_{11} + X_{11}Y_{10} + X_{10},\\
0 =&X_{11}Y_{11} + X_{10} + X_{10}Y_{12},\\
0 =&1 + Y_{10} + X_{12} + X_{11} + X_{11}Y_{12} + X_{10}Y_{12},\\
0 =&Y_{12} + Y_{10} + X_{12}Y_{10} + X_{10}Y_{12},\\
0 =&Y_{12} + Y_{10} + X_{12} + X_{12}Y_{11} + X_{10},\\
0 =&1 + Y_{11} + X_{12}Y_{12} + X_{12}Y_{11} + X_{10},\\
0 =&Y_{12} + Y_{10} + X_{10},\\
0 =&1 + Y_{12} + Y_{11} + Y_{10}Y_{12} + X_{11} + X_{10},\\
0 =&Y_{12} + Y_{11} + Y_{11}Y_{12} + Y_{10} + X_{12} + X_{10}.
\end{aligned}
$$

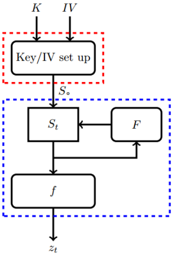
Example

if $ciphertext = (Z_{10}, Z_{11}, Z_{12}) = (1, 0, 0)$ *and* $plaintext = (Z_{00}, Z_{01}, Z_{02}) = (0, 0, 0)$ then substitute those values in relations and then solve equations. thus we have $(K_{00}, K_{01}, K_{02}) = (0, 0, 1)$.

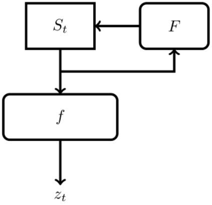
Structure of stream ciphers

The basic type of stream cipher, uses a sequence of key bits which are generated from a given initial secret key using a key generator. This keystream is then combined with the stream of plaintext bits using a XOR operation to obtain the ciphertext.





Algebrization of stream ciphers



Parameters

initial state: $S_t = (s_{t,0}, ..., s_{t,l-1})$
state function:

$$F : \mathbb{F}_2^l \to \mathbb{F}_2^l$$
$$S_t \mapsto S_{t+1} = F(S_t)$$

Next State:
$S_{t+1} = (s_{t+1,0}, s_{t+1,1}, ..., s_{t+1,l-1})$

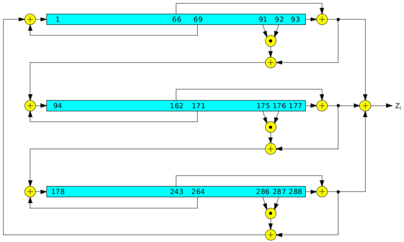
$$f : \mathbb{F}_2^l \to \{0,1\}$$
$$S_t \mapsto z_t = f(S_t)$$

Equations

$$\begin{cases} z_0 = f(s_0, s_1, ..., s_{l-1}) \\ z_1 = f(F(s_0, s_1, ..., s_{l-1})) \\ ... \\ z_{l-1} = f(F^{l-1}(s_0, s_1, ..., s_{l-1})) \end{cases}$$

Structure of Trivium

- Three nonlinear LFSRs (NLFSR) of length 93, 84, 111
- XOR-Sum of all three NLFSR outputs generates key stream z_i
- Initialization
  - load 80-bit IV into first $NLFSR$
  - load 80-bit key into second $NLFSR$
  - set $NLFSR3_{109}, NLFSR3_{110}, NLFSR3_{111} = 1$, all other bits 0
  - Clock cipher 1152 times without generating output
- For encryption, XOR-Sum of all three NLFSR outputs generates key stream z_i



Structure of Trivium

Trivium Key Generation

for $t = 1, \ldots, N$ **do**

$t_1 \leftarrow s_{66} + s_{93}$

$t_2 \leftarrow s_{162} + s_{177}$

$t_3 \leftarrow s_{243} + s_{288}$

$z_t \leftarrow t_1 + t_2 + t_3$

$t_1 \leftarrow t_1 + s_{91} \cdot s_{92} + s_{171}$

$t_2 \leftarrow t_2 + s_{175} \cdot s_{176} + s_{264}$

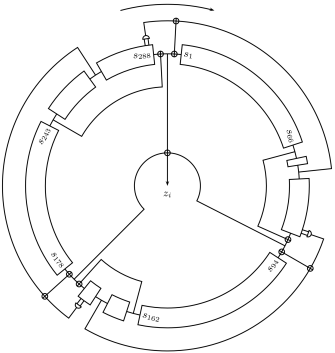
$t_3 \leftarrow t_3 + s_{286} \cdot s_{287} + s_{69}$

$(s_1, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$

$(s_{94}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$

$(s_{178}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$

end for



Algebrization of Trivium I

We can use two approach for generating equations.

- generate equations adding three new variables for any bit of keystream.

$$s_{289} = s_{66} + s_{91} \cdot s_{92} + s_{93} + s_{171}$$
$$s_{290} = s_{162} + s_{175} \cdot s_{176} + s_{177} + s_{264}$$
$$s_{291} = s_{243} + s_{286} \cdot s_{287} + s_{288} + s_{69}$$

Moreover we get one equation from the known keystream bit z_1:

$$z_1 = s_{66} + s_{93} + s_{162} + s_{177} + s_{243} + s_{288}.$$

We repeat this procedure, so at each clock we have three new variables and four equations.

Algebrization of Trivium II

- generate equations without adding any variable, to have a system which has $s_1, \ldots, s_{288}$ as unknown.

$$
\begin{aligned}
z_1 =& s_{66} + s_{93} + s_{162} + s_{177} + s_{243} \\
z_2 =& s_{65} + s_{92} + s_{161} + s_{176} + s_{287} \\
&\vdots \\
z_{66} =& s_1 + s_{28} + s_{97} + s_{112} + s_{178} + s_{223} \\
z_{67} =& s_{175} \cdot s_{176} + s_{286} \cdot s_{287} + s_{27} + s_{96} + s_{111} + s_{162} + s_{177} + s_{222} \\
& + s_{243} + s_{264} + s_{288} \\
&\vdots \\
z_{148} =& s_{13} \cdot s_{14} + s_{28} \cdot s_{29} + s_{79} \cdot s_{80} + s_{91} \cdot s_{92} + s_{94} \cdot s_{95} + s_{139} \cdot s_{140} \\
& + s_{160} \cdot s_{161} + s_{205} \cdot s_{206} + s_{232} \cdot s_{233} + s_3 + s_{30} + s_{54} + s_{66} + s_{81} \\
& + s_{93} + s_{96} + s_{108} + s_{126} + s_{141} + s_{147} + s_{159} + s_{162} + s_{171} + s_{183} \\
& + s_{189} s_{207} + s_{228} + s_{234} + s_{249}. \\
&\vdots
\end{aligned}
$$

Algebrization of public key cryptosystem

scenario 1

Write down the encryption map as a system of polynomials in the indeterminates representing the plaintext unit(s), substituting the public key and the given cipertext unit(s). Solve the polynomial system and recover the plaintext.

scenario 2

Using the public key and arbitrarily chosen plaintext units, compute the corresponding ciphertext units. Write down a polynomial representation of the decryption map using the bits of the secret key as indeterminates. Solve the polynomial system and thus find the secret key.

Algebrization of RSA I

Example

we use the RSA cryptosystem with $n = 15 = p \cdot q$, with $p = 3$ and $q = 5$. The public exponent is $e = 3$, the secret one is $d = 3$, so $de \overset{8}{\equiv} 1$ with $\phi(n) = 8$. The plaintext and ciphertext units are represented as tuple $(x_0, x_1, x_2, x_3, x_4) \in \mathbb{F}_2^4$ corresponding to elements $x_0 + 2x_1 + 4x_2 + 8x_3 \in \mathbb{Z}_{15}$. A straightforward calculation shows tht $E_{pk}(x_0, x_1, x_2, x_3) = (x_0 + 2x_1 + 4x_2 + 8x_3)^3$ is then represented by $(c_0, c_1, c_2, c_3) \in \mathbb{F}_2^4$ where

$$
\begin{aligned}
c_0 =& x_0x_1x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_1 + x_0x_2x_3 + x_0 + x_1x_2x_3 + x_2x_3\\
c_1 =& x_0x_1x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_0x_3 + x_1x_2x_3 + x_1x_2 + x_3\\
c_2 =& x_0x_1x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_1 + x_0x_2x_3 + x_1x_2x_3 + x_2x_3 + x_2\\
c_3 =& x_0x_1x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_0x_3 + x_1x_2x_3 + x_1x_2 + x_1
\end{aligned}
$$

Algebrization of RSA II

Thus we can decipher for example the ciphertext $2 = (0,1,0,0)$ bby finding the $\mathbb{F}_2$-rational solution of the polynomial system $c_0 = c_1 + 1 = c_2 = c_3 = 0$. For instance, this can be done by computing a reduced Gröbner bases of the ideal

$$J = \langle c_0, c_1 + 1, c_2, c_3, {x_0}^2 - x_0, {x_1}^2 - x_1, {x_2}^2 - x_2, {x_3}^2 - x_3 \rangle$$

computing with sagemath (`http://www.sagemath.org`):

```
sage: P.<x0,x1,x2,x3> = PolynomialRing(GF(2), order = 'lex')
sage: I = P.ideal([f0, f1 + 1, f2, f3])
sage: J = I + sage.rings.ideal.FieldIdeal(P)
sage: J.groebner_basis()
[x0, x1,x2,x3 + 1]
sage: J.variety()
[{x2: 0, x1: 0, x0: 0, x3: 1}]
```

therefore plaintext is $8 = (0,0,0,1)$.

Outline

Solver Families

In cryptography there are families of algorithms which are usually used for solving systems of equations.

1. XL, XSL, MutantXL which are based on linearization techniques.
2. Groebner basis methods: Buchberger's algorithm, F_4, F_5, ...
3. Border basis methods: BBA algorithm.
4. Mixed Integer (Linear) Programmin Solvers.
5. SAT solvers.

It is very useful to understand a bit how these solvers work.

linearization and relinearization methods I

Example

Suppose we want to solve the following system of polynomial equations in $F_7[x_1, x_2, x_3]$.

$$\begin{cases} x_1 + x_2 + x_1x_2 = 1 \\ x_2 + x_1x_2 = 1 \\ x_1 + x_1x_2 = 0 \end{cases}$$

For every product x_ix_j, we introduce a new variable y_{ij} and solve resulting *linearized* system of equations. We get $y_1 = x_1, y_2 = x_2, y_{12} = x_1x_2$ and thus

$$\begin{cases} y_1 + y_2 + y_{12} = 1 \\ y_2 + y_{12} = 1 \\ y_1 + y_{12} = 0 \end{cases}$$

$$\{y_1 = 0, y_{12} = 0, y_2 = 1\} \Longrightarrow \{x_1 = 0, x_2 = 1\}.$$

linearization and relinearization methods II

Example

Suppose we want to solve the following system of polynomial equations in $\mathbb{F}_7[x_1, x_2, x_3]$.

$$\begin{cases} 3\,{x_1}^2 + 5\,x_1\,x_2 + 5\,x_1\,x_3 + 2\,{x_2}^2 + 6\,x_2\,x_3 + 4\,{x_3}^2 = 5 \\ 6\,{x_1}^2 + x_1\,x_2 + 4\,x_1\,x_3 + 4\,{x_2}^2 + 5\,x_2\,x_3 + {x_3}^2 = 6 \\ 5\,{x_1}^2 + 2\,x_1\,x_2 + 6\,x_1\,x_3 + 2\,{x_2}^2 + 3\,x_2\,x_3 + 2\,{x_3}^2 = 5 \\ 2\,{x_1}^2 + x_1\,x_3 + 6\,{x_2}^2 + 5\,x_2\,x_3 + 5\,{x_3}^2 = 0 \\ 4\,{x_1}^2 + 6\,x_1\,x_2 + 2\,x_1\,x_3 + 5\,{x_2}^2 + x_2\,x_3 + 4\,{x_3}^2 = 0 \end{cases}$$

For every product $x_i x_j$, we introduce a new indeterminate y_{ij} and solve the resulting linearized system of equations.

$$\begin{cases} 3\,y_{11} + 5\,y_{12} + 5\,y_{13} + 2\,y_{22} + 6\,y_{23} + 4\,y_{33} = 5 \\ 6\,y_{11} + y_{12} + 4\,y_{13} + 4\,y_{22} + 5\,y_{23} + y_{33} = 6 \\ 5\,y_{11} + 2\,y_{12} + 6\,y_{13} + 2\,y_{22} + 3\,y_{23} + 2\,y_{33} = 5 \\ 2\,y_{11} + y_{13} + 6\,y_{22} + 5\,y_{23} + 5\,y_{33} = 0 \\ 4\,y_{11} + 6\,y_{12} + 2\,y_{13} + 5\,y_{22} + y_{23} + 4\,y_{33} = 0 \end{cases}$$

linearization and relinearization methods III

Example

We get $y_{11} = 2 + 5z$, $y_{12} = z$, $y_{13} = 3 + 2z$, $y_{22} = 6 + 4z$ and $y_{23} = 6 + z$, $y_{33} = 5 + 3z$ with $z \in \mathbb{F}_7$. To isolate the correct solution, we use the fundamental syzygies of the terms $x_i x_j$, namely $y_{11}y_{23} = y_{12}y_{13}, y_{12}y_{23} = y_{13}y_{22}$, and $y_{12}y_{33} = y_{13}y_{23}$ and obtain new equations for z, namely

$$3\,z^2 + z + 5 = 0\ ,\ 0z^2 + 4\,z + 4 = 0\ ,\ z^2 + 4\,z + 3 = 0$$

Now we apply a ***relinearization*** step: we introduce $z_1 = z$ and $z_2 = z^2$, solve the linear system, and find $z_1 = 6, z_2 = 1$. This yields $y_{11} = 4, y_{22} = y_{33} = 2$, and hence $x_1 = \pm 2, x_2 = \pm 3, x_3 = \pm 3$. Finally, $y_{12} = 6$ and $y_{23} = 5$ imply $(x_1, x_2, x_3) \in \{(2, 3, 4), (5, 4, 3)\}$.

XL

Based on relinearization technique, N. Courtois, A. Klimov, J. Patarin and A.Shamir proposed in the XL Algorithm (which stands for eXtended Linearization) for solving a system of multivariate quadratic equations

$$\mathcal{S} := \begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ f_2(x_1, \ldots, x_n) = 0 \\ \quad \vdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

XL Algorithm

By *XL Algorithm* we mean the procedure defined by the following steps.

1. Choose a number $D > 2$ such that $D \geq \frac{n}{\sqrt{m}}$.
2. From all products $x^{\alpha} \cdot f_i$ where $1 \leq i \leq m$ and $x^{\alpha} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is a term of *degree* $\leq D - 2$.
3. Linearize the set of all $x^{\alpha} f_i$ and solve the linear sys- tem using Gaussian elimination. The elimination has to be performed in such a way that the variables y_j corresponding to some set $\{1, x_k, ..., x_k^d\}$ are eliminate last.
4. Assume that step (3) yields at least one univariate equation $c_0 + c_1 x_k + \cdots + c_d x_k^d = 0$. Solve this equation.
5. Substitute the values of x_k back into the system and simplify it. Repeat the process to find the values of the other variables.

Using Groebner and Border bases for solving polynomial equations

Groebner bases generalize Gaussian elimination.

Example

consider the following polynomial system in $\mathbb{Q}[x, y, z]$.

$$xz^2 - 3yz + 1 = 0, \quad x^2 - 2y = 0, \quad xy - 5z = 0$$

The Gröbner basis with respect to lexicographic ordering:

$$\left.\begin{array}{l} \left.\begin{array}{l} \left.\begin{array}{l}\{x^2 - 5x + 6,\end{array}\right\} x \text{ only} \\ 2xy - 4y - x + 2, \\ 4y^2 + 3x - 10, \end{array}\right\} x, y \text{ only} \\ xz^2 - 2z^2 - 4x + 8, \\ z^3 - 4xz + 8z + 2x - 6\}. \end{array}\right\} \text{all variables}$$

The solutions of such a system can be easily determined.

Theorem

Let I be a zero dimentional ideal of $K[x_1, ..., x_n]$ and σ be a monomial ordering on $\mathbb{T}^n$, and let $\mathcal{O}_\sigma(I)$ be the order ideal $\mathbb{T}^n \setminus \mathrm{LM}_\sigma(I)$. Then there exists a unique $\mathcal{O}_\sigma(I)$-border basis G of I, and the reduced σ-Groebner basis of I is the subset of G.

Mixed Integer Programming I

Let $f_1, ..., f_m \in P = \mathbb{F}_2[x_1, ..., x_n]$. Then the following instructions defines a zero of the 0-dimensional radical ideal $I = \langle f_1, ..., f_m, x_1^2 + x_1, ..., x_n^2 + x_n \rangle$.

1. Reduce $f_1, ..., f_m$ module the field equations, i.e. make their support square-free. For $i = 1, ..., m$, let S_i be the set of terms of *degree* ≥ 2 in f_i and $s_i = |\operatorname{Supp}(f_i)|$.
2. For $i = 1, ..., m$, introduce aa new indeterminate k_i and write down the linear inequality $K_i : k_i \leq [\frac{s_i}{2}]$.
3. For every $t_j \in S_i$, introduce a new indeterminate y_{ij}. For $i = 1, ..., m$, write $f_i = \sum_j t_j + l_i$ where the sum extends over all j such that $t_j \in S_i$ and where $l_i \in P_{\leq 1}$. Form the linear equation $F_i : \sum_j y_{ij} + l_i - 2k_i = 0$.
4. For $i \in \{1, ..., m\}$ and $t_j \in S_i$, write $t_j = x_{j_1} ... x_{j_r}$ with $1 \leq j_1 < \cdots < j_r \leq n$. Form the linear inequalities $Y_{ij} : y_{ij} - x_i \leq 0$ and $Z_{ij} : -y_{ij} + x_{j_1} + \cdots + x_{j_r} - r + 1 \leq 0$.
5. For all $i \in \{1, ..., m\}$, let $X_i : x_i \leq 1$.

Mixed Integer Programming II

6. Choose a linear polynomial $C \in \mathbb{Q}[x_i, y_{ij}, k_i]$ and use an IP solver to find the tuple of natural numbers (a_i, b_{ij}, c_i) which solves the system of linear equations and inequalities $\{K_i, F_i, Y_{ij}, Z_{ij}, X_i\}$ and minimizes C.
7. Return $(a_1, ..., a_n)$ and stop.

Comparing the size and the time of solving CTC(B,N)'s equations with `GBasis5(...)` command of CoCoA and with GLPK package applied to the IP problem of previous algorithm (laptop with 2.0 GHz processor and 2GB of RAM)

CTC(b,N)	n	m	t	time GBasis5	time GLPK	sol. unique?
CTC(2,2)	54	98	60	0.3 s	0.2 s	yes
CTC(2,3)	78	144	90	2.0 s	1.0 s	yes
CTC(3,2)	81	147	90	2.8 s	2.0 s	yes
CTC(3,3)	117	216	135	∞	12.7 s	yes
CTC(3,4)	153	285	180	∞	85.8 s	no

Questions?

Thank You!

97.87 %

References I

- Gregory V. Bard, Nicolas T. Courtois, and Chris Jefferson.
  Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers.
  Cryptology ePrint Archive, Report 2007/024, 2007.
- Nicolas T. Courtois, Gregory V. Bard and David Wagner.
  Algebraic and Slide Attacks on KeeLoq
  In *Fast Software Encryption – FSE 2008*, pages 97–115, Berlin, Heidelberg, New York, 2008. Springer Verlag
- Nicolas T. Courtois.
  Security Evaluation of GOST 28147-89 In View Of International Standardisation,
  In *Cryptology ePrint Archive*, Report 2011/211, 2011
- Julia Borghoff, Lars R. Knudsen, and Mathias Stolpe.
  Bivium as a Mixed-Integer Linear programming problem.
  In Matthew G. Parker, editor, *Cryptography and Coding – 12th IMA International Conference*, volume 5921 of *Lecture Notes in Computer Science*, pages 133–152, Berlin, Heidelberg, New York, 2009. Springer Verlag.

References II

Useful websites:

- SageMath: `http://www.sagemath.org`
- N. Courtois:
  `http://nicolascourtois.com`
  `http://blog.bettercrypto.com`
- M. Albrecht:
  `https://martinralbrecht.wordpress.com`
  `https://bitbucket.org/malb`