MODELLING THE RC4 CIPHER AS AN INTEGER QUADRATIC PROGRAMMING PROBLEM

Members:
Ng Wen Xi (Nanyang Girls' High School)
Teo Woo Hng Reanne (Eunoia Junior College)
Esther Keziah Lim Rui Qi (Methodist Girls' School)

Mentor:
Ruth Ng Ii Yung, Lim Zhan Feng
(DSO National Laboratories)

BACKGROUND

Rivest Cipher 4 (RC4)

- Wide application in network protocols
  - Speed and simplicity
  - Easy to implement
  - Used in SSL, TLS, WEP, WPA and Microsoft Windows
- Development of many attacks
  - e.g. Bias Attack [1]

MOTIVATION

- RC4 still used in WEP and Microsoft Windows up to date [2]
- GOAL: Prove the insecurity of RC4 with new forms of mathematical analysis
- TOOL: Integer Quadratic Programming Solver
  - Efficient
  - Second least complex class in the hierarchy of convex optimisation problems

NOVEL CONTRIBUTIONS

- We expressed the Key Scheduling Algorithm (KSA), one of the two algorithms making up RC4, as mathematical expressions
- We compiled and devised linearisation techniques for all five nonlinear KSA operators. (bitwise AND and OR, mod, a class of piecewise function, multiplication of a binary and nonbinary variable.)
- We consolidate these techniques, producing a fully linear expression of KSA
- Using this, we were able to model the entire KSA as an IQP problem

What is an Integer Quadratic Programming Problem?

Inputs:

1. Quadratic Objective function of the following form:

Function is maximised or minimised

$$f(x_0,\dots,x_{n-1}) = \sum_{i,j=0}^{n-1} \underbrace{a_{i,j}}_{\text{Coefficient}} x_i x_j + \underbrace{b_i}_{\text{Coefficient}} x_i + \underbrace{c}_{\text{Constant}}$$

n is the number of variables

$x_0, x_1, x_2 \dots, x_{n-1}$ *are integer variables*

2. A set of linear constraints with the following form:

$$\sum_{i=0}^{n-1} \underbrace{d_i}_{\text{Coefficient}} x_i + \underbrace{e}_{\text{Constant}} \le 0$$

Output:

1. $(t_0,\dots,t_{n-1})$, where $(t_0,\dots,t_{n-1})$ are the values of the variables $(x_0,\dots,x_{n-1})$ when the objective function is optimised and all constraints are satisfied

Example:

$f(x_0,x_1,x_2) = x_0^2 + x_1^2$ — Objective Function

$0 \le x_0 \le 1$
$x_1 + x_2 = 3$ — Constraints

Solver

$(t_0,t_1,t_2) = (0,0,3)$

What is Key Scheduling Algorithm?

$element^{\,loop\ number\ where\ term\ is\ first\ introduced}_{\,placing\ of\ element\ in\ tuple\ for\ each\ loop\ number}$

'SWAP'

Each s variable in the new loop is assigned to the variable introduced in the previous loop with the same subscript.
Exception occurs when subscript is i or j^{i+1}
When the variable introduced in the new loop has subscript i, it will be assigned the value of the variable introduced in the previous loop with subscript j^{i+1} and vice versa.

Key Scheduling Algorithm (n,k)

$j^0 \leftarrow 0$

$\underbrace{s_0^0, s_1^0, \dots, s_{n-1}^0}_{\text{Initial State}} \leftarrow 0, 1, \dots, \underbrace{n}_{\text{Integer between 1 and 256}}\text{-}1$

For $i = 0, \dots, n-1$

$j^{i+1} \leftarrow (j^i + s_i^i + k^i) \bmod n$

SWAP:
$s_0^{i+1}, s_1^{i+1}, s_2^{i+1}, \dots, s_{n-1}^{i+1} \leftarrow s_0^i, s_1^i, s_2^i, \dots, s_{n-1}^i$
$s_i^{i+1} \leftarrow s_{j^{i+1}}^i$
$s_{j^{i+1}}^{i+1} \leftarrow s_i^i$

Return $\underline{s}$ State (tuple of length n(n+1))

Methodology (How to use Integer Quadratic Programming for Key Scheduling Algorithm?)

GOAL:
'known-plaintext attack' to derive key when initial and final state is known. Final state is given while Initial state is standard.

Inputs:

1. Objective Function:

$$\sum_{m=0}^{n-1} (fs_m - s_m^n)^2 = (fs_0 - s_0^n)^2 + (fs_1 - s_1^n)^2 + \dots + (fs_{n-1} - s_{n-1}^n)^2$$

(Known) Final State: $fs_0, fs_1, \dots, fs_{n-1}$; Derived Final State: $s_0^n, s_1^n, \dots, s_{n-1}^n$

The objective function is to be minimised.

2. KSA Algorithm

Output:

1. **Key, k**

$(k^0, k^1, \dots, k^{n-1})$

Problem: Solvers for IQP problems do not accept KSA as pseudocode as constraints

1. Not valid mathematical equations
2. Nonlinear

How to get KSA to be expressed as Linear Constraints?

1. Express KSA Algorithm as a set of equations
2. Find techniques to replace nonlinear mathematical operators used in equations with linear constraints
3. Apply techniques to the set of equations

Expressing KSA as a set of constraints

From the pseudo code, all arrows can be converted to equality signs. This is with the exception for the 3 line 'SWAP'. Why?

1. Mathematical expression would be non valid; two different values to the same variable.
E.g. If $i = 0$ and $j^{i+1} = 1$, $s_0^1 = s_0^0$ and $s_0^1 = s_1^0$
2. Solver is unable to solve when there are variables in the subscript of the terms.
E.g. $s_i^{i+1} = s_{j^{i+1}}^i = s_{(j^i + s_i^i + k^i) \bmod n}^i$

α: $s_0^{i+1}, s_1^{i+1}, s_2^{i+1}, \dots, s_{n-1}^{i+1}$
$\leftarrow s_0^i, s_1^i, s_2^i, \dots, s_{n-1}^i$: β
$s_i^{i+1} \leftarrow s_{j^{i+1}}^i$
$s_{j^{i+1}}^{i+1} \leftarrow s_i^i$

For $\beta = 0, \dots, n-1$

$$s_\beta^{i+1} = \sum_{\alpha=0}^{n-1} a_{\alpha,\beta}^{i+1} s_\alpha^i = a_{0,\beta}^{i+1} s_0^i + a_{1,\beta}^{i+1} s_1^i + \dots + a_{n-1,\beta}^{i+1} s_{n-1}^i$$

where $a_{\alpha,\beta}^{i+1}$ *are* ***decision variables***

If **any one** of the following main criteria are met, $a_{\alpha,\beta}^{i+1}$=1, else $a_{\alpha,\beta}^{i+1}$=0
1. When $\alpha = i$ *and* $\beta = j^{i+1}$
2. When $\alpha = j^{i+1}$ *and* $\beta = i$
3. When $\alpha = \beta$, $\alpha \ne i$ *and* $\alpha \ne j^{i+1}$

Let $u_{\alpha,\beta,1}^{i+1}$, $u_{\alpha,\beta,2}^{i+1}$ *and* $u_{\alpha,\beta,3}^{i+1}$ be variables that indicate if each of the criteria are being fulfilled.
E.g. If criteria 1 is being fulfilled, $u_{\alpha,\beta,1}^{i+1} = 1$ else, $u_{\alpha,\beta,1}^{i+1} = 0$

Thus, $a_{\alpha,\beta}^{i+1} = u_{\alpha,\beta,1}^{i+1} \lor u_{\alpha,\beta,2}^{i+1} \lor u_{\alpha,\beta,3}^{i+1}$

Sub criteria
1. $\alpha = i$ 2. $\beta = j^{i+1}$
3. $\alpha = j^{i+1}$ 4. $\beta = i$
5. $\alpha = \beta$ 6. $\alpha \ne i$ 7. $\alpha \ne j^{i+1}$

Let $v_{\alpha,\beta,1}^{i+1}, v_{\alpha,\beta,2}^{i+1}, v_{\alpha,\beta,3}^{i+1}, v_{\alpha,\beta,4}^{i+1}, v_{\alpha,\beta,5}^{i+1}$ be variables that indicate if each of the sub criteria 1-5 are fulfilled.

E.g. $v_{\alpha,\beta,1}^{i+1} = \begin{cases} 1 \text{ when } \alpha = i \\ 0 \text{ when } \alpha \ne i \end{cases}$

Thus,
$$u_{\alpha,\beta,1}^{i+1} = v_{\alpha,\beta,1}^{i+1} \land v_{\alpha,\beta,2}^{i+1}$$
$$u_{\alpha,\beta,2}^{i+1} = v_{\alpha,\beta,3}^{i+1} \land v_{\alpha,\beta,4}^{i+1}$$
$$u_{\alpha,\beta,3}^{i+1} = v_{\alpha,\beta,5}^{i+1} \land (1 - v_{\alpha,\beta,1}^{i+1}) \land (1 - v_{\alpha,\beta,3}^{i+1})$$

Linearising mathematical operators

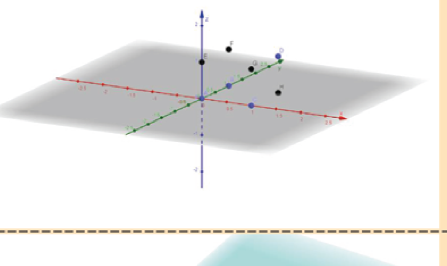
Bitwise operators

- **Bitwise AND ($\land$)**

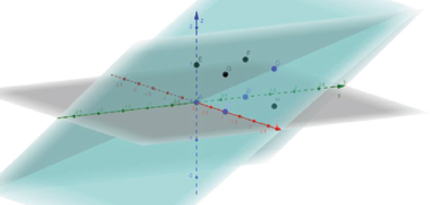
Goal: Derive a set of linear constraints that forces z to be equal to $x \land y$, given $x, y = 0$ *or* 1.

First, we set $0 \le z \le 1$ and $z \in \mathbb{Z}$

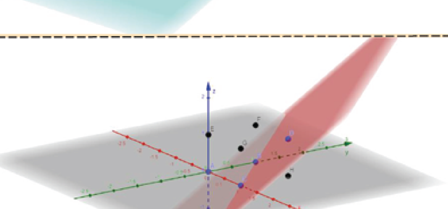
x	y	$x \land y = z$
0	0	0
0	1	0
1	0	0
1	1	1



There are now 8 possible sets of values x, y and z can take. $x \land y = z$ only for points in blue (A,B,C,D)



Points E, F, G lie above one of the blue planes. To exclude Points E,F,G, we use constraints: $z \le y$ and $z \le x$



Only point H lie under the red plane. To exclude Point H, we use the constraint: $z \ge x + y - 1$

For example, when $x = 1, y = 0, z = x \land y = xy = 0$
Using constraints, $0 \le z \le 1$
$z \in \mathbb{Z}$
$z \le 1$ and $z \le 0$
$z \ge 1 + 0 - 1$
Thus, $z = 0$

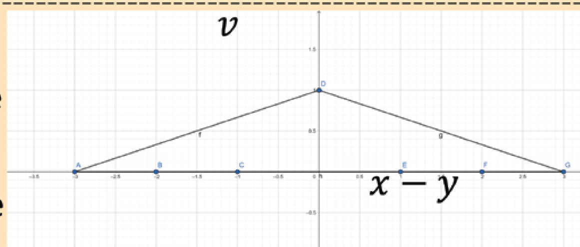
RC4 Piecewise Function

Goal: Derive a set of linear constraints that forces z to be equal to either 0 or 1 according to the following form of piecewise function:

$$v = \begin{cases} 1 \text{ when } x = y \\ 0 \text{ when } x \ne y \end{cases}$$
where $0 \le x \le g, 0 \le y \le g$ *and* $x, y \in \mathbb{Z}$

$$v = \begin{cases} 1 \text{ when } x - y = 0 \\ 0 \text{ when } x - y \ne 0 \end{cases}$$
where $-g \le x - y \le g$ *and* $x, y \in \mathbb{Z}$

Let g=3
All possible points (A to G) are plotted.



A, D, G expressed in terms of g are A($-g$,0), D(0,1), G(g,0). Thus, we have the constraints:

$v \le 1 - \frac{x-y}{g}$ (Point is under line AD)
$v \le 1 + \frac{x+y}{g}$ (Point is under line DG)
$v \ge 0$ (Point is above line AG)
$v \in \mathbb{Z}$

Ambiguous Case: When $x - y = 0$, $\boldsymbol{v = 0 \text{ or } 1}$

Linearising mod

When $a = b \bmod n$, a is the remainder of $\frac{b}{n}$

$a = b - xn$
$0 \le a \le n - 1$
$x \in \mathbb{Z}$

Forces xn to be highest multiple of n while $0 \le a$

Conclusion

In this work, we managed to successfully model KSA of the RC4 cipher into a convex IQP problem. We overcame the challenge of KSA not fulfilling the structure of an IQP problem, by expressing it as mathematical expressions before linearising the constituting operators. Thus, we proved that given the final state, an attacker would be able to derive the key.

Other work done

We linearised bitwise OR as well as the multiplication of a binary and nonbinary variable using similar methods. We were then able to write out a general form of a fully linear expression of KSA and thus a general form for modelling KSA as an IQP problem.

References
[1] I. Mantin, A. Shamir
I.Martin, A.Shamir. 2001. A Practical Attack on Broadcast RC4. https://www.scribbr.com/citation/generator/folders/w2KoEozEWOZ33ajic8uaa/lists/4zZWbGrBSVXJImn2awEUgF/
[2] Sarkar, S., Sen Gupta, S., Paul, G., & Maitra, S. 2014. Proving TLS-attack related open biases of RC4 https://dl.acm.org/doi/10.1007/s10623-014-0003-0