第八届（2023）全国高校密码数学挑战赛

赛题一

一、赛题名称：有限域密码原语 CICO 问题

二、赛题描述

2.1 符号说明

- p：素数
- $\mathbb{F}_p$：p元素域
- $\mathrm{mod}\, n$：模n取余数
- $X = (X_0, X_1, \cdots, X_{t-1})$：由$\mathbb{F}_p$上$t$个元素$X_0, X_1, \cdots, X_{t-1}$构成的状态$X$, 其中$X_i$表示从左向右第$i$个元素，$0 \le i \le t-1$
- $M[i,j]$：矩阵M第i行、第j列的元素，指标i、j均从 0 开始

2.2 基础知识

$\mathbb{F}_p$上的加法运算为整数加法模p，乘法运算为整数乘法模p。例如在$\mathbb{F}_7$上，$3+6=2, 3\times 6=4$。给定函数$F: \mathbb{F}_p^t \to \mathbb{F}_p^t$，CICO（Constrained-Input Constrained-Output）问题描述如下：

对于给定的$(a_1, \cdots, a_k), (b_1, \cdots, b_k) \in \mathbb{F}_p^k$，其中 $2k < t$，问题的目标是找到$(X_1, \cdots, X_{t-k})$和$(Y_1, \cdots, Y_{t-k})$，使得

$$F(X_1, \cdots, X_{t-k}, a_1, \cdots, a_k) = (Y_1, \cdots, Y_{t-k,} b_1, \cdots, b_k).$$

2.3 赛题描述

设 $p = 18446744073709551557$，$F: \mathbb{F}_p^3 \to \mathbb{F}_p^3$，。本赛题的目标为找到$P_0, P_1, C_0, C_1 \in \mathbb{F}_p$，使得

$$F(P_0, P_1, 0) = (C_0, C_1, 0).$$

参赛者需要：

(1) 设计求取P_0, P_1, C_0, C_1的方案，给出方案和理论复杂度评估，要求理论复杂度小于p次域操作；

(2) 实际求解出的对应P_0, P_1, C_0, C_1。

R轮迭代函数F表示为

$$F = F^{(R)} \circ F^{(R-1)} \circ \cdots \circ F^{(1)}.$$

输入为$P = (P_0, P_1, P_2)$，输出为$C = (C_0, C_1, C_2)$。第r轮迭代$F^{(r)}$如图 1 所示，$1 \le r \le R$。其中，输入表示为 $X^{(r)} = (X_0^{(r)}, X_1^{(r)}, X_2^{(r)})$。$F^{(r)}$由两步组成，每步包括非线性函数$S_i$、线性函数$L$和常数加$AC$三部分组成，$i = 1,2$。其中，第一步使用非线性函数$S_1$，第二步使用非线性函数$S_2$，即

$$F^{(r)} = (AC \circ L \circ S_2) \circ (AC \circ L \circ S_1)$$

同时，在第 1 轮轮函数前还有一次额外的常数加操作。

(1) 非线性函数S：记非线性函数S的输入为(X_0, X_1, X_2)，输出为(Y_0, Y_1, Y_2)，$F^{(r)}$第 1 步所使用到的非线性函数S_1表示为

$$Y_0 = (X_0)^3, Y_1 = (X_1)^{1/3}, Y_2 = (X_2)^3.$$

第 2 步所使用到的非线性函数S_2表示为

$$Y_0 = (X_0)^{1/3}, Y_1 = (X_1)^3, Y_2 = (X_2)^{1/3}.$$

(2)线性函数L：记线性函数L的输入为(X_0, X_1, X_2)，输出为(Y_0, Y_1, Y_2)，则$F^{(r)}$使用的线性函数表示为

$$(Y_0, Y_1, Y_2)^T = M \cdot (X_0, X_1, X_2)^T.$$

其中，M为$\mathbb{F}_p$上3×3矩阵，T表示转置。

(3)常数加AC：$F^{(r)}$中，两步使用到的轮常数依次为$Con^{((2r-2) \bmod 10)}$与$Con^{((2r-1) \bmod 10)}$，在第 1 轮轮函数前使用到的轮常数为$Con^{(-1)}$，其中，

$$Con^{(i)} = (Con_0^{(i)}, Con_1^{(i)}, Con_2^{(i)})$$

矩阵M、轮常数具体取值和测试向量参见附件。

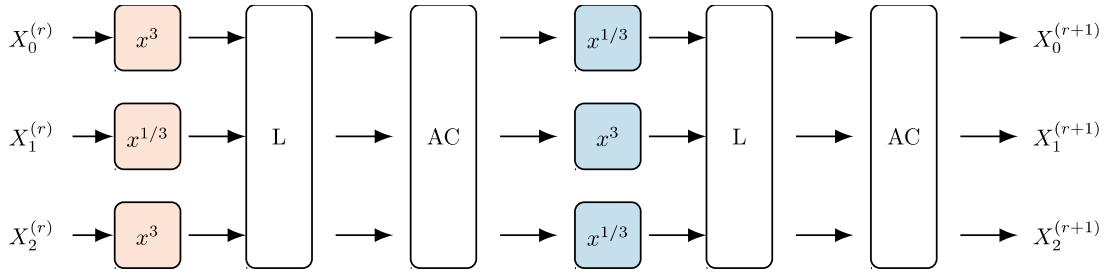

图 1：函数F第r轮迭代$F^{(r)}$

函数F伪代码如下：

输入：输入$P = (P_0, P_1, P_2)$，迭代次数R

输出：输出$C = (C_0, C_1, C_2) = F(P_0, P_1, P_2)$

for $i = 0$ **to** 2 **do**

$X_i^{(0)} = P_i + Con_i^{(-1)}$

for $r = 1$ **to** R **do**

$Y_0^{(r-1)} = \left(X_0^{(r-1)}\right)^3, Y_1^{(r-1)} = \left(X_1^{(r-1)}\right)^{1/3}, Y_2^{(r-1)} = \left(X_2^{(r-1)}\right)^3$

for $i = 0$ **to** 2 **do**

$Z_i^{(r-1)} = Con_i^{((2r-2)\bmod 10)} + \sum_{j=0}^{2} M[i,j] \times Y_j^{(r-1)}$

end

$U_0^{(r-1)} = \left(Z_0^{(r-1)}\right)^{1/3}, U_1^{(r-1)} = \left(Z_1^{(r-1)}\right)^3, U_2^{(r-1)} = \left(Z_2^{(r-1)}\right)^{1/3},$

for $i = 0$ **to** 2 **do**

$X_i^{(r)} = Con_i^{((2r-1)\bmod 10)} + \sum_{j=0}^{2} M[i,j] \times U_j^{(r-1)}$

end

end

return $(X_0^{(R)}, X_1^{(R)}, X_2^{(R)})$

2.4 成绩评判标准

评判时，将对函数F每步求解情况进行评分，总步数为$2R$步，各步得分之和即为选手最后的总得分。总得分计算公式为

$$\delta = \sum_{s=1}^{2R} \delta_s, \qquad \delta_s = t_s + p_s$$

δ_s为第s步总得分，由理论分数t_s和实验分数p_s组成。其中，$t_s = 0.3 \times \delta_s$，$p_s = 0.7 \times \delta_s$。各步总分分别为：

$$\delta_s = \begin{cases} 100 \times s, 1 \le s \le 4 \\ 120 \times s, 5 \le s \le 10 \\ 150 \times s, s > 10 \end{cases}$$

比赛排名规则：

(1) 最终总分最高者获胜，分数相同的队伍比较其在最高步数中的得

分，如分数仍相同，则比较次高步数得分，以此类推。

(2) 如分数无法分出胜负，则根据完成挑战的方法（如求取方案、理论复杂度等）进行综合评定。

提交结果说明：

(1) 参赛者在报告中需要详细描述求取方案并给出复杂度评估，引用前人的方法需要在报告中明确指出，否则结果作废；

(2) 针对实际求解出的挑战实例，需给出实验环境、计算所需时间和内存、源程序及可执行代码和计算结果；

(3) 利用特殊方法求解或求解方法中有创新内容的，可以酌情加分。

三、密码学背景及相关问题的研究进展

近年来，随着安全多方计算、全同态加密、零知识证明等隐私计算技术的落地应用，涌现出许多隐私计算友好的对称加密原语。例如，在零知识证明协议中需要使用这类新型对称原语构造的哈希函数（通常使用海绵结构构造），从而基于 Merkle 树验证成员身份。证明协议的性能也高度依赖于该哈希函数描述中使用到的算术操作数目。由于零知识证明协议通常部署在有限域$\mathbb{F}_p$上（p一般取大素数），因此考虑到性能优化，这类新型对称加密原语也定义在$\mathbb{F}_p$上。

同时，零知识证明协议的安全性也高度依赖于使用的哈希函数的抗碰撞性。为了衡量新型对称原语在海绵结构下的安全性，我们引入 CICO（Constrained-Input Constrained-Output）问题：

令函数$F:\mathbb{F}_p^t \to \mathbb{F}_p^t$，对于给定的$(a_1,\cdots,a_k),(b_1,\cdots,b_k) \in \mathbb{F}_p^k$，$2k < t$，问题的目标是找到$(X_1,\cdots,X_{t-k})$和$(Y_1,\cdots,Y_{t-k})$，使得

$$F(X_1,\cdots,X_{t-k},a_1,\cdots,a_k) = (Y_1,\cdots,Y_{t-k},b_1,\cdots,b_k).$$

CICO 问题与哈希函数的安全性高度相关。因此，如果敌手有能力以小于p^k次置换调用的复杂度求解，就有可能找到海绵结构下哈希函数的原像或碰撞。

CICO 问题通常可以建模成方程系统并利用代数手段进行求解。目前被广泛认可的求解方程系统的方法主要有因式分解、Gröbner 基算法等方法[1,2]。对于多变元高次方程系统来说，Gröbner 基算法是目前被广泛认可的有效算法，利用 Gröbner 基算法对密码算法进行攻击的步骤如下：

(1) 将密码算法建模成多变元方程系统，不同的建模方式会导致不同的攻击复杂度;

(2) 利用 F4/F5 算法计算多变元方程系统的 Gröbner 基；

(3) 使用 FGLM 算法将计算出的 Gröbner 基转化成字典序 Gröbner 基;

(4) 利用单变元方程求解等技巧计算出方程的最终解。

Gröbner 基攻击的复杂度主要由第 2 步和第 3 步决定。在第 2 步中，复杂度定义为

$$\mathcal{O}\binom{D_{reg}+n_v}{n_v}^{\omega}$$

次操作，其中D_{reg}为正则度，n_v表示变量数目，$2 \le \omega < 3$为矩阵乘法的复杂度成熟。同时，令n_e表示方程数目，d_i表示每个方程的次数。当方程系统为正则系统时，即$n_e = n_v$，正则度表示为$D_{reg} = 1 +$

$\sum_{i=1}^{n_e}(d_i-1)$。当方程为半正则系统时，正则度可以估计为方程

$$H(z)=\frac{\prod_{i=1}^{n_e}(1-z^{d_i})}{(1-z)^{n_v}}$$

的第一个非正系数。

在第 3 步中，复杂度定义为

$$\mathcal{O}(n_v\cdot d^3)$$

其中，d为理想$\mathcal{I}=\langle P_1,P_2\cdots,P_{n_e}\rangle$的次数，$P_1,P_2\cdots,P_{n_e}$为建模得到的$n_e$个方程。

四、参考文献

[1] Sauer J F, Szepieniec A. Sok: Gröbner basis algorithms for arithmetization oriented ciphers[J]. Cryptology ePrint Archive, 2021.

[2] Albrecht M R, Cid C, Grassi L, et al. Algebraic cryptanalysis of STARK-friendly designs: application to MARVELlous and MiMC[C]//Advances in Cryptology–ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part III 25. Springer International Publishing, 2019: 371-397.