Solving Modular Linear Equations via Automated Coppersmith and its Applications

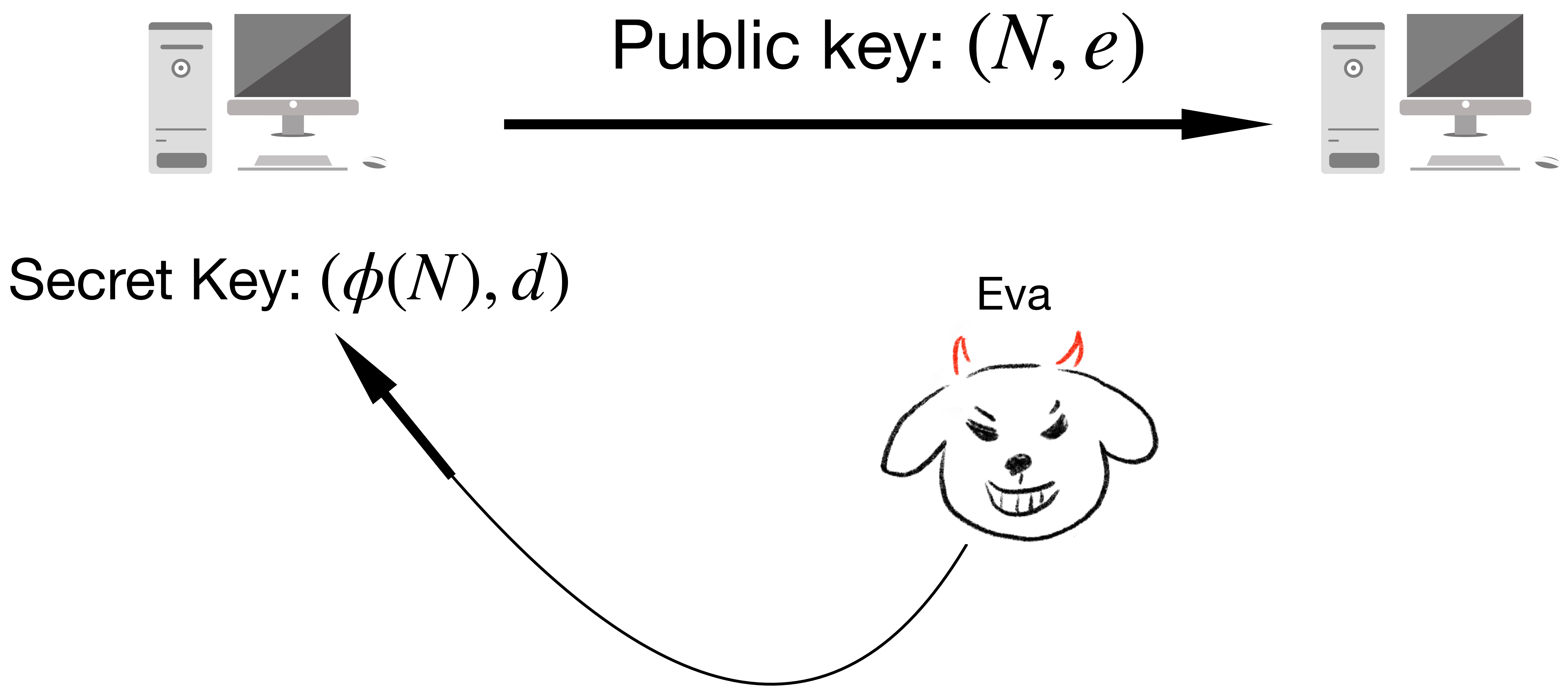
Yansong Feng   Zhen Liu   Abderrahmane Nitaj   Yanbin Pan

- Background
- Lattice-based Cryptanalysis: Coppersmith’s method
- Implicit Factorization Problem

Background

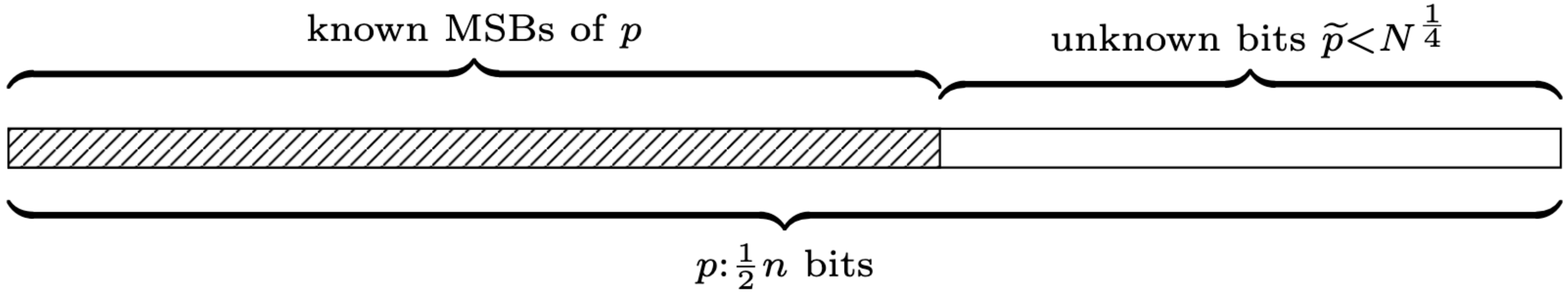
RSA Cryptosystem

$ed \equiv 1 \mod \phi(N) \quad \phi(N) = (p-1)(q-1)$



Eve wants to get the SECRET KEY!!!

Lucky Eva got enough MSBs of p…



Now he just needs to solve a linear polynomial equation:

$f(x) = x + C \equiv 0 \mod p$ with a small root $x_0 = \widetilde{p} < N^{\frac{1}{4}}$

How to solve polynomials equations with small roots?

Coppersmith’s method

Coppersmith's method

Let $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n} \in \mathbb{R}^m$, the lattice $\mathscr{L}$ is

$$\mathscr{L} = \left\{ \mathbf{v} \in \mathbb{R}^m \mid \mathbf{v} = \sum_{i=1}^{n} a_i \mathbf{v_i}, a_i \in \mathbb{Z} \right\}.$$

Give bound X_j and $f \in \mathbb{Z}[x_1, \ldots, x_k]$ and modulus M, the goal is to find the small root $\mathbf{u} = (u_1, \ldots, u_k)$ with $u_j < X_j$, such that $f(\mathbf{u}) \equiv 0 \mod M$.

1. Use the coefficient vector of $g_j(x_1X_1, \ldots, x_kX_k)$ to construct $\mathscr{L}$

2. Using Lattice Reduction find Shorter vectors $\cancel{h_1, \ldots, h_k}$

$$h_j(\mathbf{u}) \equiv 0 \mod M \longrightarrow h_j(\mathbf{u}) = 0$$

$\mathscr{L}$ MUST satisfied $\det(\mathscr{L}) < M^{m \dim(\mathscr{L})}$.

$$\det(\mathscr{L}) < p^{\frac{1}{2}m\dim(\mathscr{L})}$$

$f(x) = x + C \equiv 0 \mod p$ with a small root $x_0 = \widetilde{p} < N^{\frac{1}{4}}$

N^4	0	0	0	0	0	0	0	0
$*$	N^3X	0	0	0	0	0	0	0
$*$	$*$	N^2X^2	0	0	0	0	0	0
$*$	$*$	$*$	NX^3	0	0	0	0	0
$*$	$*$	$*$	$*$	X^4	0	0	0	0
$*$	$*$	$*$	$*$	$*$	X^5	0	0	0
$*$	$*$	$*$	$*$	$*$	$*$	X^6	0	0
$*$	$*$	$*$	$*$	$*$	$*$	$*$	X^7	0
$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	X^8

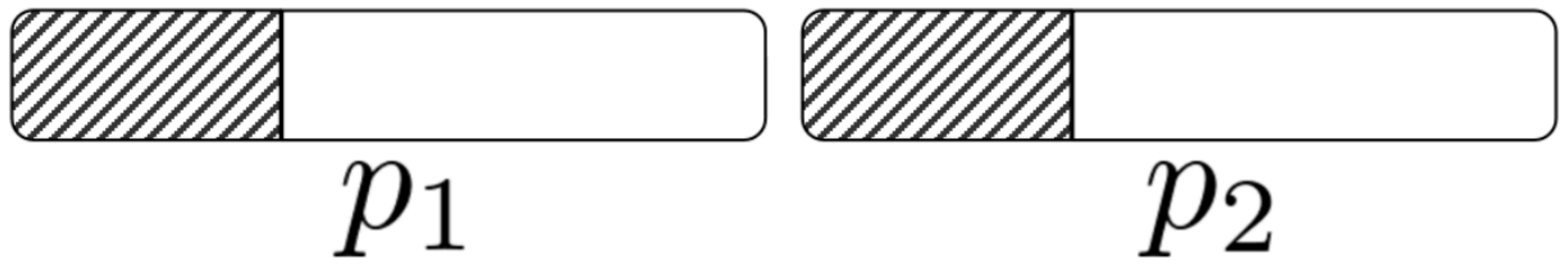
$$\dim(\mathscr{L}) = m + o(m)$$

$$\det(\mathscr{L}) = N^{\frac{1}{8}m^2+o(m^2)}X^{\frac{1}{2}m^2+o(m^2)}$$

$X < N^{\frac{1}{4}} \rightarrow \det(\mathscr{L}) < p^{m\dim(\mathscr{L})} \rightarrow f$ can be solved with Coppersmith's method.

Implicit Factorization Problem

IFP (MSBs case)



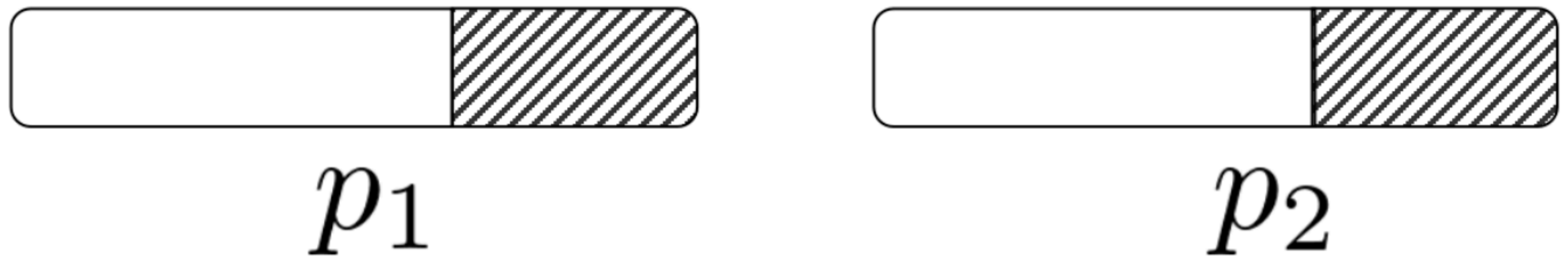
$N_1 = p_1q_1$ and $N_2 = p_2q_2$

p_1 share the same MSBs with p_2

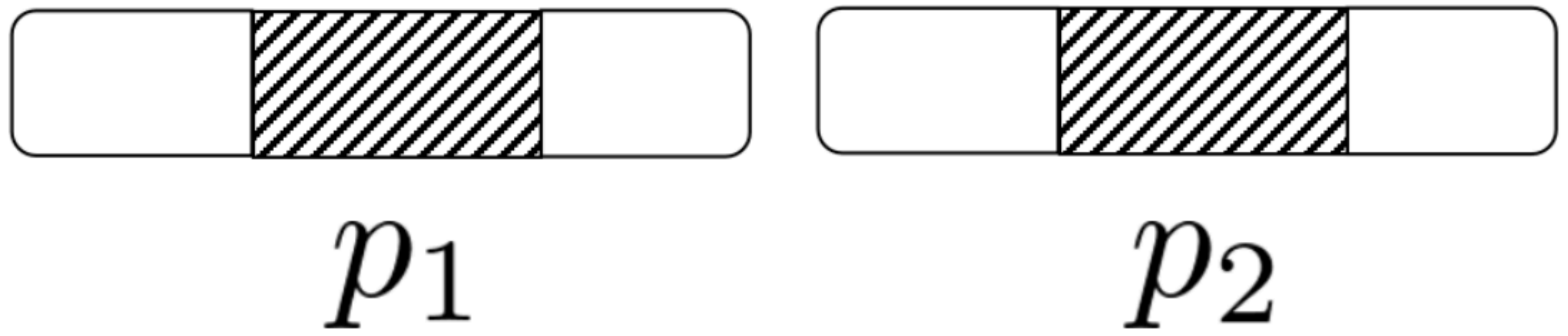
$$N_1 + (p_2 - p_1)q_1 = p_2q_1 \equiv 0 \mod p_2$$

Solving $f(x_1, x_2) = x_1x_2 + N_1 \equiv 0 \mod p_2$ with $(p_2 - p_1, q_1)$

IFP (LSBs case)



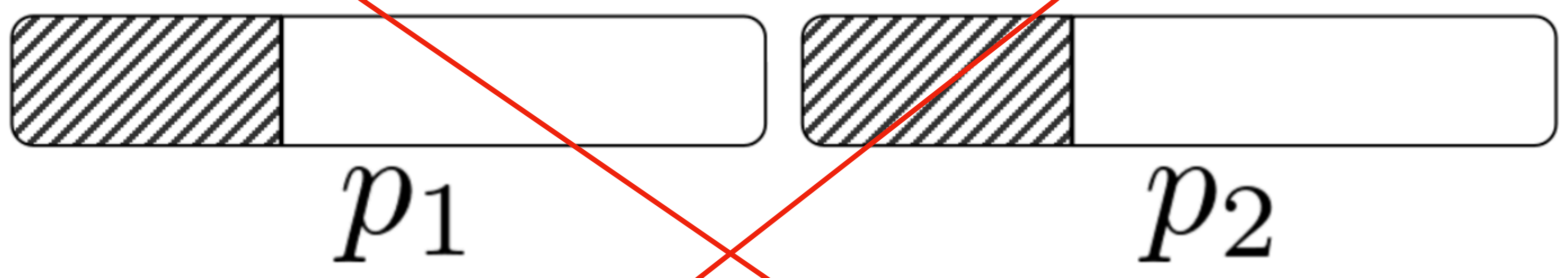
IFP (Middle case)



IFP (Generalized case)

(a) p_1 (b) p_2

EIFP (MSBs case) $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$



~~p_1 share the same MSBs with p_2~~ $\longrightarrow$ $a_1 p_1$ share the same MSBs with $a_2 p_2$

How about EIFP with Generalized case? G-EIFP!

$a_1 p_1$ share some continuous bits with $a_2 p_2$, which can be located in different positions.

Suppose that a_1p_1 and a_2p_2 share γn-bits so that

$$a_1p_1 = x_1 + 2^{\beta_1 n}R + x_2 2^{(\beta_1+\gamma)n},$$
$$a_2p_2 = x_3 + 2^{\beta_2 n}R + x_4 2^{(\beta_2+\gamma)n},$$

$f(x, y, z) = x + ay + N_2 z$ with $(x_0, y_0, z_0) = \left(2^{(\beta_2-\beta_1)n}x_1q_2 - x_3q_2, x_2q_2 - x_4q_2, a_2\right)$.

Using Coppersmith's method, compute $\dim(\mathscr{L})$ and $\det(\mathscr{L})$:

Manual Calculation such as calculating $\sum_{k=0}^{m}\sum_{i=k}^{m}(i - \min(s, i))$? NO!

Theorem: $\dim(\mathscr{L})$ and $\det(\mathscr{L})$ are polynomials in m.

Now,

~~Manual Calculation~~ $\longrightarrow$ Lagrange Interpolation

(s, m_i)	(0, 0)	(0, 1)	(1, 1)	(0, 2)	(1, 2)	(2, 2)	(0, 3)	(1, 3)	(2, 3)	(3, 3)
$p_w(s, m_i)$	0	1	0	4	1	0	10	4	1	0
$p_v(s, m_i)$	0	0	2	0	3	8	0	4	11	20

Lagrange Interpolation $\longrightarrow$

$$
\begin{aligned}
p_x &= \frac{1}{6}m^3 + o\left(m^3\right), \\
p_y &= \frac{1}{6}m^3 + o\left(m^3\right), \\
p_z &= \frac{1}{6}m^3 + o\left(m^3\right), \\
p_w &= \frac{1}{6}(1-\tau_2)^3 m^3 + o\left(m^3\right), \\
p_v &= \frac{1}{6}\left(-\tau_2^3 + 3\tau_2^2\right) m^3 + o\left(m^3\right), \\
p_{\mathcal{F}_1} &= \frac{1}{6}\tau_1^2(3-\tau_1) m^3 + o\left(m^3\right), \\
p_{\mathcal{F}_2} &= \frac{1}{3}m^3 + o\left(m^3\right), \\
p_{\mathcal{M}} &= m|\mathcal{M}| = \frac{1}{2}m^3 + o\left(m^3\right).
\end{aligned}
$$

$\dim(\mathscr{L})$ and $\det(\mathscr{L})$

$$n : \log_2 N_i, \quad \alpha : \log_2 q_i, \quad \delta : \log_2 a_i, \quad \gamma : \frac{\text{shared bits}}{n}$$

Theorem: $\text{G-EIFP}(n, \alpha, \gamma, \delta)$ can be solved in polynomial time when

$$\gamma > 4\alpha\left(1 - \sqrt{\alpha}\right) + 2\delta,$$

provided that $\alpha + \gamma \leq 1$.

Thanks for listening!

Code: https://github.com/fffmath/CombeeIFP