Cryptanalysis

Sasha

June 30, 2020

1 Information Set Decoding

[**?**](just to not modify makefile added some references)

Notation: we take a vector $t \in \mathbb{Z}^k$, a matrix $A \in \mathbb{Z}^{k \times n}$. It can be reduced to its(maybe permuted) systematic form $H = U * A = [I_k|D]$.

The goal is to find a vector $x \in \{0,1\}^n$ with small Hamming weight $h(x) = w$ that

$$Ax = y$$

Or equivalently $Hx = U * Ax = Uy =: t$, here we can use our knowledge about the shape of H.

What we do in this attack is an improvement of a bruteforce attack. The partition x on two vectors $x_1 \in \{0,1\}^k$ and $x_2 \in \{0,1\}^{n-k}$ so we have

$$t = x_1 + D \cdot x_2$$

We make a bet of the weight partition between $h(x_1) = w_1$ and $h(x_2) = w_2$, where $w_1 + w_2 = w$. Now we enumerate only the possible values of x_2, compute $x_1 = t - D \cdot x_2$ and check if it satisfies $h(x_1) = w_1$. If we don't find a correct pair with this weight distribution, we rerandomise H and t and start over. The average cost of such algorithm can be calculated as

$$T = \frac{x_2 \text{ bruteforce cost}}{Pr(w_2 \text{ is a correct bet on the weight of } x_2)}$$

Let us compute the values above. The numerator:

$$\#\{x_2 \in \{0,1\}^{n-k} | h(x_2) = w_2\} = \binom{n-k}{w_2}$$

The denominator:

$$\begin{aligned} Pr(w_2 \text{ is a correct bet on the weight of } x_2) & \\ &= Pr(h(x_2) = w_2 | h(x) = w) \\ &= \frac{\binom{n-k}{w_2} \cdot \binom{k}{w_1}}{\binom{n}{w}} \end{aligned}$$

Therefore

$$T = \frac{\binom{n}{w}}{\binom{k}{w_1}}$$

Lemma 1. *To minimize the average cost we take* $w_1 = min(\frac{k}{2}, w)$

Proof. □

1.1 Ternary case

2 Meet in the Middle

In this attack we have the same goal but no information about the form of the matrix A. We partition x and A on two equal parts: $A = [A_1|A_2]$, $x = x_1|x_2$. Then $Ax = y$ is equivalent to

$$A_1x_1 + A_2x_2 = y$$

If we can find vectors x_1 and x_2 for which values A_1x_1 and $y - A_2x_2$ coincide and sum of their weights is equal to w they form a solution to our problem.

Here we bet the weight is distributed equaly on both sides. So the average cost can be calculated as follows:

$$T = \frac{\text{cost of finding a collision}}{Pr(h(x_1) = h(x_2) = w/2 | h(x) = w)}$$

Numerator: We compute A_1x_1 for every x_1 and store it in the memory. So we perform $\binom{n/2}{w/2}$ operations the same amount of memory. In the worst case we also compute $y - A_2x_2$ for every x_2 without storing it - $\binom{n/2}{w/2}$ operations.

In total:

TIME $= 2\binom{n/2}{w/2}$

MEMORY $= \binom{n/2}{w/2}$

Denominator: $Pr(h(x_1) = h(x_2) = w/2) = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$

Total cost(only time):

$$T = \frac{2\binom{n}{w}}{\binom{n/2}{w/2}}$$

Question 1. *how do we rerandomise in this case? $\rightarrow$ We can just multiply by any unimodular matrix!*

2.1 Ternary case

3 ISD + MiM

Let us return to the case when A is reduced to the systematic form $H = U{*}A = [I_k|D_1|D_2]$ we partition $x = (x_0|x_1|x_2)$ on three vectors $x_0 =\in \{0,1\}^k$, $x_1, x_2 \in \{0,1\}^{\frac{n-k}{2}}$. Then

$$Ax = x_0 + D_1x_1 + D_2x_2$$

We make bet that $h(x_0) = w_1$, $h(x_1) = h(x_2) = \frac{w_2}{2}$ and perform Meet-in-the-Middle attack trying to find a collision between D_2x_2 and all possible $D_1x_1 + x_0$

For that store a table of D_2x_2 in memory and look-up there for $D_1x_1 + x_0$.

Comutational cost:

$$\left(1 + \binom{k}{w_1}\right)\binom{\frac{n-k}{2}}{\frac{w_2}{2}}$$

Memory cost:

$$\binom{\frac{n-k}{2}}{\frac{w_2}{2}}$$

Total cost:

$$\begin{aligned} T &= \frac{\text{cost of collision search}}{Pr(h(x_0) = w_1, h(x_1) = h(x_2) = \frac{w_2}{2}|h(x) = w)} \\ &= \frac{(1 + \binom{k}{w_1})\binom{\frac{n-k}{2}}{\frac{w_2}{2}} \cdot \binom{n}{w}}{\binom{k}{w_1} \cdot \binom{(n-k)/2}{w_2/2}^2} \\ &\sim \frac{\binom{n}{w}}{\binom{(n-k)/2}{w_2/2}} \end{aligned}$$

4 Question 5 and 6