Randomized Iterative Affine Cipher

在secureboost中，需要对一阶微分/二阶微分结果进行大量的加密和解密，基于已有的Paillier[1]在2048以及更高的安全位下面，性能非常的差。FATE从新提出一种基于新的支持加法同态的加密体系：随机迭代仿射密码。

普通的仿射密码[2] 原理非常简单：

假设$(a, n) = 1$:

- 加密： $E(x) = ax + b(mod\ n)$
- 解密： $D(x) = a^{-1}(x - b)(mod\ n)$

正确性显而易见。

安全性上面， 很容易通过频率分析找到仿射对应关系。单纯的仿射密码并不安全。下面我们看看RIAC是怎么防止这个问题的。

RIAC

RIAC在仿射的基础上，进行多轮同余乘法运算, 乘法对加法满足分配率，进而实现同态加法。

密码系统介绍

先看下该密码系统：

- 秘钥生成：

  input: key_size, key_round, encode_precision

  output: a_array, n_array, g, x， 其中a_array是私钥。

  algo:

  ```
  1. 对于0<=i<key_round， 随机生成key_size位的n_array[i], 然后找出相同位数的互质的a_array[i]; 注意实际实现的时候为了防止彩虹表攻击，这里使用了不等长的key_size.
  2.  生成长度为key_size//10的g, 以及160bit的x
  ```

  encode_precision的用途后面会提到。

- 加密

$$Enc\_i(x) = a_i x + b (mod\ n_i)$$
$$Enc(x) = Enc\_n * \ldots * Enc\_1(x)$$

其中$[*]$ 表示模乘，也就是同余乘法。

- 解密

$$Dec\_i(x) = a_i^{-1}(x - b) (mod\ n_i)$$
$$Enc(x) = Dec\_n * \ldots * Dec\_1(x)$$

- 同态加法

$$Enc(x_1) + Enc(x_2) = Enc\_n * \ldots * Enc\_1(x_1) + Enc\_n * \ldots * Enc\_1(x_2)$$
$$= Enc\_n * \ldots * Enc\_1(x_1 + x_2)$$

可以看到，同态加法和加解密一样只需要模乘，相对Paillier在加解密的时候逆运算，性能要好很多。

对浮点数的支持

浮点数在多次运算过程中会有损失，一般的同态会考虑借助bootstrapping[3]等手段进行噪音控制。RIAC的实现思路就很简单，只需要暴力的将a_i 扩大2**100倍即可。同时因为secureboost多用于金融模型，考虑可解释性，行业中一般将树的深度设置为2。因此这种放大倍数完全可以保证精度不受损失。

安全分析

单纯的仿射密码基于频率分析较为容易破解，但是结合多轮递归仿射，每一轮的仿射都采取了不同长度的秘钥。

对于单轮的仿射： a是一个key_size长度的数字，那么可能数个数是2^{key_size} 。通过key_round=r轮，那么所有可能的数个数为$2^{\sum_r key_size_i} + \beta$。其中$\beta$是额外加入的仿射因子， 默认是2 ** 23。

实现

见这里.

Reference

1. Paillier实现： https://github.com/xuperdata/teesdk/tree/master/paillier
2. Affine cipher: https://en.wikipedia.org/wiki/Affine_cipher
3. Gentry, Craig, Shai Halevi, and Nigel P. Smart. "Better bootstrapping in fully homomorphic encryption." *International Workshop on Public Key Cryptography*. Springer, Berlin, Heidelberg, 2012.