

解决组合公钥共谋攻击和密钥碰撞的新方法*

李方伟, 马安君, 朱江, 余航

(重庆邮电大学 移动通信技术重庆市重点实验室, 重庆 400065)

摘要: 以解决组合公钥体制中共谋攻击和密钥碰撞问题为目的。首先, 针对线性共谋攻击, 提出了一种新的构造种子矩阵的方法, 使得种子密钥和大于基点加法群的阶数, 从而使密钥之间不能相互线性表示。其次在密钥的生产过程中, 引入系数破坏了层不同和层互斥不同的关系, 为解决选择共谋攻击提供了一种有效的方法, 同时增强了抵御随机共谋攻击的能力。最后, 在密钥产生的流程中, 通过公钥对比来避免密钥碰撞, 为解决密钥碰撞问题提出了一种新方法。

关键词: 组合公钥; 共谋攻击; 密钥碰撞; 线性共谋攻击; 选择共谋攻击

中图分类号: TP309 **文献标志码:** A **文章编号:** 1001-3695(2014)04-1176-04

doi:10.3969/j.issn.1001-3695.2014.04.053

New method to solve collusion attack and key collision in combined public key

LI Fang-wei, MA An-jun, ZHU Jiang, YU Hang

(Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts & Telecommunications, Chongqing 400065, China)

Abstract: The purpose of this paper is to solve the problem of collusion attack and key collision in combined public key cryptosystem. First, for the linear collusion attacks, this paper proposed a new method to construct seed matrix, it made the sum of seeds greater than the order of addition group generated by base point. Secondly, in the production process of key, the introduction of different coefficients destroyed the type layer and the layer mutex type relationship, it provided an effective way for solving chosen collusion attack, and enhanced the ability against random collusion attack. Thirdly, comparing public keys was used to avoid key collision in the key production process, and proposed a new method to solve key collision.

Key words: combined public key (CPK); collusion attack; key collision; liner collusion attack; chosen collusion attack

1 概述

组合公钥(CPK)体制^[1-3]由我国学者南湘浩教授于1999年提出。CPK算法^[4]依据椭圆曲线离散对数问题的数学原理构建公钥种子矩阵和私钥种子矩阵,采用hash运算和映射算法将用户的身份标志映射为矩阵的行、列坐标,用于对矩阵元素进行选取与组合,生成数量庞大的公私密钥对。CPK具有大规模密钥管理功能,不需要可信的第三方进行在线认证,可以实现离线身份认证、多域和跨域认证等。与之相比,目前使用最广泛的公钥基础设施(public key infrastructure, PKI)需要通过可信权威机构认证中心(certification authority, CA)生成的CA证书来进行身份确认,即身份确认过程必须建立在对第三方共同信任的基础之上。基于身份的密码体制(identity based encryption, IBE)^[5](身份即公钥),不需要证书和相应的认证过程及相关的开销,受到越来越多的关注。但是由于私钥产生器(private key generator, PKG)拥有主密钥,生成所有用户的私钥,能够解密加密方案中的任何密文,能够伪造任何用户的签名,因此私钥托管成为基于身份的密码体制的难题^[6]。无证书密码体制^[7]和基于身份系统一样不需要公钥证书。同时,由于密钥生成中心(key generation center, KGC)只生产和

传递部分私钥,有效地避开了基于身份系统中的私钥托管问题。可以说,无证书公钥密码系统很好地结合了PKI和IBE的优点,但是每次通信之前都要进行公钥验证,增加了一定的开销。

由于CPK用户私钥是由用户身份标志通过hash运算、映射算法、模加运算而产生,私钥间存在一定的关系,因此存在三种共谋攻击,即线性共谋攻击^[8,9]、选择共谋攻击^[10]、随机共谋攻击^[10]。线性共谋攻击是最基本的共谋攻击,一直以来提出了一些针对私钥进行保护^[8]的方案,从而避免共谋攻击的方法,但并没有实质性地解决线性共谋攻击问题。文献[11,12]提出了双矩阵组合公钥算法,解决了选择共谋攻击和随机共谋攻击问题,同时一定程度上提高了抵抗线性共谋攻击的能力,但并没有完全解决线性共谋攻击问题,同时也没有综合考虑到密钥碰撞的问题。

密钥碰撞是组合公钥体制的另一个问题。文献[13]分析了密钥碰撞的概率;文献[14]提出了种子矩阵的优化方法,避免了种子和相等的情形;文献[15]在文献[14]的基础上提出了分离矩阵列数量级的方法解决了种子和相等情形,同时通过约束椭圆曲线参数的方法解决了种子和模加相等的情形;文献[16]用分组密码对用户标志进行映射,并按照特定规则产生

收稿日期: 2013-05-29; 修回日期: 2013-07-15 基金项目: 国家自然科学基金资助项目(61071116,61271260)

作者简介: 李方伟(1960-),男,重庆人,教授,博士,主要研究方向为移动通信技术与理论、组网技术、信息安全技术、信号处理和智能天线技术等(lifw@cqupt.edu.cn); 马安君(1986-),男,重庆万州人,硕士研究生,主要研究方向为加密、数字签名、认证机制; 朱江(1977-),男,湖北荆州人,副教授,博士,主要研究方向为认知无线电技术; 余航(1988-),女,四川泸州人,硕士研究生,主要研究方向为移动通信、信息安全。

种子公私钥库,优化后种子库规模小、构建效率高、占用空间少。以上解决密钥碰撞的方法均在一定程度上解决了部分密钥碰撞的问题,但没有结合共谋攻击综合考虑 CPK 体制的安全问题。本文从密钥产生的流程出发,为解决密钥碰撞的问题提出了一种新的密钥产生的方案。

2 组合公钥算法

CPK 算法建立在椭圆曲线密码(elliptic curve cryptography, ECC)系统上,理论依据是椭圆曲线离散对数问题(elliptic curve discrete logarithm problem, ECDLP)的难解性。其参数记为 $T = (a, b, G, n, p)$ 。其中 a, b 是定义在有限域 p 上的椭圆曲线 $y^2 \equiv (x^3 + ax + b) \pmod p$ 的非负整数; G 是加法群的基点; n 是以 G 为基点的群的阶; p 是有限域的阶,是一个很大的素数。

2.1 密钥矩阵的生成

在给定椭圆曲线参数 T 的基础上,由离线的密钥管理中心构建密钥矩阵。密钥矩阵大小为 $m \times h$,公钥矩阵 PSK 的元素记为 $R_{i,j}$,私钥矩阵 SSK 的元素记为 $r_{i,j}$ 。本算法中密钥矩阵构建方式如下:

a) 系统初始化,在有限域 F_p 上,以 $T = \{a, b, G, n, p\}$ 为参数构建椭圆曲线。

b) 生成随机数 $r_{i,j}$,验证 $r_{i,j}$ 是否满足 $\frac{n}{h} < r_{i,j} < n$,不满足则继续执行 b)。

c) 根据 $R_{i,j} = r_{i,j}G = (x_{i,j}, y_{i,j})$,计算公钥矩阵元素 $(x_{i,j}, y_{i,j})$ 。

d) 检查矩阵是否生成完成,否则执行 b)。

$$PSK = \begin{bmatrix} (x_{1,1}, y_{1,1}) & (x_{1,2}, y_{1,2}) & \cdots & (x_{1,h}, y_{1,h}) \\ (x_{2,1}, y_{2,1}) & (x_{2,2}, y_{2,2}) & \cdots & (x_{2,h}, y_{2,h}) \\ \vdots & \vdots & & \vdots \\ (x_{m,1}, y_{m,1}) & (x_{m,2}, y_{m,2}) & \cdots & (x_{m,h}, y_{m,h}) \end{bmatrix} \quad (1)$$

$$SSK = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,h} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,h} \\ \vdots & \vdots & & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,h} \end{bmatrix} \quad (2)$$

PSK 与 SSK 矩阵对应的元素 $R_{i,j}$ 和 $r_{i,j}$ 构成公私密钥对。ECC 具有复合特性:任意多对私钥之和与对应的公钥之和构成新的公私密钥对。

密钥管理中心保留私钥矩阵、公钥矩阵、系统参数 $T = \{a, b, G, n, p\}$,以及其他相关参数,公开公钥矩阵、系统参数和其他相关参数。

2.2 密钥的生成

原 CPK 算法基于身份标志的密钥的产生是通过对标志进行 hash 运算和行映射算法实现的。首先采用无碰撞的 hash 函数将标志变换为固定长度的中间变量 $data$,然后行映射算法采用分组密码算法,以 Rowkey 为密钥进行循环式加密生成行坐标的随机数序列,即 $MAP_{i+1} = E_{ROWKEY}(MAP_i)$,其中 $MAP_0 = data$,列坐标则按顺序使用。行坐标序列与列坐标序列唯一确定与标志对应的组合密钥的构成元素。

本文算法中,对选取的元素乘以系数然后进行模加运算构成用户的私钥。设 (a_1, a_2, \dots, a_t) 为系数序列,其中 $a_i (1 \leq i \leq t)$ 为 k bit 整数,满足 $1 \leq a_1, a_2, \dots, a_t < n$,系数序列随公钥

矩阵一起公开。系数的选取方式与行坐标一样,首先采用无碰撞的 hash 函数将标志变换为固定长度的中间变量,然后采用分组密码算法以 Coekey 为密钥进行循环式加密生成序号 $i (1 \leq i \leq t)$ 。

设标志映射值对密钥矩阵的行列坐标为 $(i_1, 1), (i_2, 2), \dots, (i_h, h)$,系数序列为 (a_1, a_2, \dots, a_h) ,则私钥 sk 为

$$sk = (a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h}) \pmod n \quad (3)$$

用户公钥 PK 为

$$\begin{aligned} PK &= a_1 R_{i_1,1} + a_2 R_{i_2,2} + \cdots + a_h R_{i_h,h} = \\ & a_1 r_{i_1,1} G + a_2 r_{i_2,2} G + \cdots + a_h r_{i_h,h} G = \\ & (a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h}) G = sk \cdot G \end{aligned} \quad (4)$$

由式(4)可知,公私钥之间的关系与原 CPK 算法一致,在改进的 CPK 算法中,加解密和签名验证算法与原 CPK 算法相同,这里不作赘述。

3 共谋攻击

由于用户私钥是通过 hash 运算、映射算法、模加运算生成的,因此用户的私钥间存在一定的线性关系,根据此线性关系,多个用户共谋可以得到其他用户私钥。

3.1 线性共谋攻击

线性共谋攻击分为两种:一是通过多个用户共谋列方程组解方程得到私钥矩阵,从而得到任何用户的私钥;二是通过对多个用户私钥进行线性组合可以得到其他用户私钥。

3.1.1 线性共谋攻击原理

根据原 CPK 算法,私钥是从私钥矩阵选取 h 个元素进行模加运算产生的,如式(5)所示。

$$sk = (r_{i_1,1} + r_{i_2,2} + \cdots + r_{i_h,h}) \pmod n \quad (5)$$

其中: $r_{i_j,j} (1 \leq j \leq h)$ 是从私钥矩阵每一列所取的数。设 SSK 每一列的取值行向量为 (t_1, t_2, \dots, t_m) ,则

$$t_1 + t_2 + \cdots + t_m = 1, t_k \in \{0, 1\}, 1 \leq k \leq m \quad (6)$$

即每一列必须且只取一个数,被取到对应位置的 t_k 就等于 1,其余位置为 0。那么私钥 sk 可以表示为

$$sk = (t_1 \ t_2 \ \cdots \ t_m)_1 \begin{pmatrix} r_{1,1} \\ r_{2,1} \\ \vdots \\ r_{m,1} \end{pmatrix} + \cdots + (t_1 \ t_2 \ \cdots \ t_m)_h \begin{pmatrix} r_{1,h} \\ r_{2,h} \\ \vdots \\ r_{m,h} \end{pmatrix} \quad (7)$$

也可表示为 $sk = t \cdot r$,其中:

$$t = (t_{1,1} \ t_{1,2} \ \cdots \ t_{1,h} \ \cdots \ t_{m,1} \ t_{m,2} \ \cdots \ t_{m,h}) \quad (8)$$

$$r = (r_{1,1} \ r_{1,2} \ \cdots \ r_{1,h} \ \cdots \ r_{m,1} \ r_{m,2} \ \cdots \ r_{m,h})^T \quad (9)$$

根据私钥产生的规则可知,总共可以产生 m^h 个不同私钥。那么根据全部私钥列出方程组如下:

$$\begin{pmatrix} t_{1(1,1)} & \cdots & t_{1(1,h)} & \cdots & t_{1(m,1)} & \cdots & t_{1(m,h)} \\ t_{2(1,1)} & \cdots & t_{2(1,h)} & \cdots & t_{2(m,1)} & \cdots & t_{2(m,h)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ t_{m^h(1,1)} & \cdots & t_{m^h(1,h)} & \cdots & t_{m^h(m,1)} & \cdots & t_{m^h(m,h)} \end{pmatrix} \begin{pmatrix} r_{1,1} \\ r_{1,2} \\ \vdots \\ r_{m,h} \end{pmatrix} = \begin{pmatrix} sk_1 \\ sk_2 \\ \vdots \\ sk_{m^h} \end{pmatrix} \quad (10)$$

将系数矩阵记为 A ,元素记为 $t_{a(b,c)}$ 。其中, a 表示第几个方程; b 表示取值元素在私钥矩阵的第几行; c 表示取值元素在

私钥矩阵的第 j 列; $t_{a(b,c)}$ 表示元素的取值情况, 1 表示被取到, 0 表示未被取到。系数矩阵 A 为

$$A = \begin{pmatrix} t_{1(1,1)} & \cdots & t_{1(1,h)} & \cdots & t_{1(m,1)} & \cdots & t_{1(m,h)} \\ t_{2(1,1)} & \cdots & t_{2(1,h)} & \cdots & t_{2(m,1)} & \cdots & t_{2(m,h)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ t_{mh(1,1)} & \cdots & t_{mh(1,h)} & \cdots & t_{mh(m,1)} & \cdots & t_{mh(m,h)} \end{pmatrix} \quad (11)$$

对 A 进行线性变换, $A \xrightarrow{c_i + c_{h+i} + c_{2h+i} + \cdots + c_{(m-1)h+i}} A_1$, $1 \leq i \leq h$, 将第 $h+i, 2h+i, \dots, (m-1)h+i$ 列加到第 i 列, 可得矩阵的前 h 列元素全为 1, 那么由此可得系数矩阵 A 的秩 $R(A) \leq mh - h + 1$, 进一步可以证明系数矩阵的秩为 $mh - h + 1$ 。

3.1.2 抗线性共谋攻击

从改进的 CPK 算法可知, 私钥的计算方法有所变化。SSK 取值的行向量 (t_1, t_2, \dots, t_m) 不满足式 (6), 而是等于从随机序列中选取的数。那么系数矩阵 A 中的元素 $t_{a(b,c)}$ 也不再是非 0 即 1, 每一行中有 h 个非 0 数, 它们都是根据身份标志从系数序列中选取的随机数。由于生成行坐标所采用的 hash 函数与生成系数序号的 hash 函数不同, 则选取私钥矩阵中同一个数所对应的系数不同。对系数矩阵 A 进行如上的线性变换 $A \xrightarrow{c_i + c_{h+i} + c_{2h+i} + \cdots + c_{(m-1)h+i}} A_1$, $1 \leq i \leq h$, 得到的矩阵前 h 列元素不再是有规律的全为 1, 而是毫无规律的随机数, 无法相消, 此时系数矩阵的秩 $R(A) = mh$, 满秩。

由构造矩阵的要求可知, 每个元素满足 $\frac{n}{h} < r_{i,j} < n, 1 \leq i \leq m, 1 \leq j \leq h$, 则 $n < r_{i_1,1} + r_{i_2,2} + \cdots + r_{i_h,h} < hn$, 那么可以得到 $n < a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h}$, 即

$$sk = (a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h}) \bmod n \neq a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h} \quad (12)$$

实际上根据私钥构造的方程为

$$\begin{pmatrix} t_{1(1,1)} & \cdots & t_{1(1,h)} & \cdots & t_{1(m,1)} & \cdots & t_{1(m,h)} \\ t_{2(1,1)} & \cdots & t_{2(1,h)} & \cdots & t_{2(m,1)} & \cdots & t_{2(m,h)} \\ \vdots & & \vdots & & \vdots & & \vdots \\ t_{mh(1,1)} & \cdots & t_{mh(1,h)} & \cdots & t_{mh(m,1)} & \cdots & t_{mh(m,h)} \end{pmatrix} \begin{pmatrix} r_{1,1} \\ r_{1,2} \\ \vdots \\ r_{m,h} \end{pmatrix} = \begin{pmatrix} sk_1 + k_1 n \\ sk_2 + k_2 n \\ \vdots \\ sk_m + k_m n \end{pmatrix} \quad (13)$$

其中: k_1, k_2, \dots, k_m 为大于或等于 1 的未知整数。对增广矩阵

$$\tilde{A} = \begin{pmatrix} t_{1(1,1)} & \cdots & t_{1(1,h)} & \cdots & t_{1(m,1)} & \cdots & t_{1(m,h)} & sk_1 \\ t_{2(1,1)} & \cdots & t_{2(1,h)} & \cdots & t_{2(m,1)} & \cdots & t_{2(m,h)} & sk_2 \\ \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ t_{mh(1,1)} & \cdots & t_{mh(1,h)} & \cdots & t_{mh(m,1)} & \cdots & t_{mh(m,h)} & sk_m \end{pmatrix} \quad (14)$$

作线性变换可得 $R(\tilde{A}) = R(A) + 1$ 。由线性方程组解的相关定理可知, 当系数矩阵的秩 $R(A)$ 与其增广矩阵的秩 $R(\tilde{A})$ 不相等时, 则方程组无解。因此, 第一种线性共谋攻击方式不可行。

改进的 CPK 算法所列方程组系数矩阵 A 的秩为 mh , 说明系数矩阵 A 的 m^h 个行向量中只有 mh 个行向量线性无关, 其他行向量可以由这 mh 个行向量线性表示出来。然而系数矩阵 A 的增广矩阵 \tilde{A} 的每一个行向量才表示组成一个私钥的方程。其他全部的私钥能不能由这 mh 个线性无关的私钥表示

出来, 就要把它们表示成方程组, 考查方程组系数矩阵的秩是否与增广矩阵的秩相等。增广矩阵的秩并不等于系数矩阵的秩, 则其他所有的私钥都不可以由这 mh 个线性无关的线性表示。同时, 由于每一个私钥 $sk \neq a_1 r_{i_1,1} + a_2 r_{i_2,2} + \cdots + a_h r_{i_h,h}$, 所以任意多个用户之间进行线性运算, 其系数矩阵和增广矩阵的秩都不会相等。因此, 第二种线性共谋攻击方式也不可行。

3.2 选择共谋攻击

原 CPK 算法的存在选择共谋攻击: 设用户 A 与用户 B 是 $(j_1, j_2, \dots, j_{i_1})$ 层不同, 用户 A 与用户 C 是 $(s_1, s_2, \dots, s_{i_2})$ 层, 且集合 $\{j_1, j_2, \dots, j_{i_1}\}$ 和 $\{s_1, s_2, \dots, s_{i_2}\}$ 交集为空, 则用户 A、B、C 共谋可得到与 A 是 $\{j_1, j_2, \dots, j_{i_1}, s_1, s_2, \dots, s_{i_2}\}$ 层不同的用户的私钥。

选择共谋攻击成功的前提是私钥之间存在层不同和层互斥不同, 而本算法打破这种关系。用户 A、B、C 欲共谋得到用户 D 的私钥, 假设用户 A、B、C 满足上述层不同和层互斥不同关系, 私钥分别为

$$sk_A = a_{01} r_{0(i_1,1)} + a_{02} r_{0(i_2,2)} + \cdots + a_{0h} r_{0(i_h,h)} \quad (15)$$

$$sk_B = a_{11} r_{1(i_1,1)} + a_{12} r_{1(i_2,2)} + \cdots + a_{1h} r_{1(i_h,h)} \quad (16)$$

$$sk_C = a_{21} r_{2(i_1,1)} + a_{22} r_{2(i_2,2)} + \cdots + a_{2h} r_{2(i_h,h)} \quad (17)$$

那么根据选择共谋攻击的原理可得

$$sk_B + sk_C - sk_A = a_{11} r_{1(i_1,1)} + a_{12} r_{1(i_2,2)} + \cdots + a_{1h} r_{1(i_h,h)} + a_{21} r_{2(i_1,1)} + a_{22} r_{2(i_2,2)} + \cdots + a_{2h} r_{2(i_h,h)} - (a_{01} r_{0(i_1,1)} + a_{02} r_{0(i_2,2)} + \cdots + a_{0h} r_{0(i_h,h)}) \quad (18)$$

按照设想用户 D 的私钥应该为

$$sk_D = r_{1(i_1,1)} + r_{1(i_2,2)} + \cdots + r_{1(i_1,t_1)} + r_{2(i_1+1,t_1+1)} + r_{2(i_1+2,t_1+2)} + \cdots + r_{2(i_1+t_1,t_1+t_2)} + r_{0(i_1+t_2+1,t_1+t_2+1)} + r_{0(i_1+2,t_1+2)} + \cdots + r_{0(i_h,h)} \quad (19)$$

显然, $sk_B + sk_C - sk_A \neq sk_D$, 因此三个用户共谋得不到与 A 层 $\{j_1, j_2, \dots, j_{i_1}, s_1, s_2, \dots, s_{i_2}\}$ 不同的用户 D 的私钥, 那更不能推广到更多用户的选择共谋攻击。所以, 本算法可以抵御选择共谋攻击。

3.3 随机共谋攻击

随机共谋攻击的原理如下:

a) 两个合谋用户 A 和用户 B 计算其组合私钥的差 $\Delta sk_{BA} = sk_B - sk_A$ 和 $\Delta sk_{AB} = sk_A - sk_B$, 以及对应的公钥差 $\Delta PK_{BA} = PK_B - PK_A$ 和 $\Delta PK_{AB} = PK_A - PK_B$ 。

b) 在公钥因子矩阵中任意取得一个组合公钥 C, 如果 $PK_C - PK_B = \Delta PK_{BA}$ 或 $PK_C - PK_A = \Delta PK_{AB}$, 则该组合公钥对应的私钥为 $sk_C = 2sk_B - sk_A$ 或 $sk_C = 2sk_A - sk_B$, 否则继续执行步骤 b)。

随机共谋攻击的根本在于公私钥之间存在关系 $PK = sk \cdot G$ 。本文所提出的方法并没有破坏这种关系, 因此无法解决随机共谋攻击。但是由于密钥不再只是私钥种子模加的结果, 而是私钥种子乘以系数后模加的结果, 使得密钥的产生变得复杂, 已经不能如文献 [10] 举例的那样, 根据公钥矩阵元素层不同和私钥种子层不同的等差关系来说明公钥和私钥存在等差关系, 即已经不能单纯地根据公钥矩阵元素之间的关系来推测私钥之间的关系, 这增加了随机共谋的难度, 提高了 CPK 算法的安全性。

4 密钥碰撞

密钥碰撞就是密钥相同, 即不同的身份标志依据密钥生成

原理产生了相同的私钥和公钥。密钥碰撞可能引起身份识别错误、签名伪造、通信的保密性和不可否认性丧失等一系列问题。

4.1 密钥碰撞的原因

由私钥生成过程可知,如图1所示,排除身份标志相同的可能,密钥碰撞的原因有两个,即映射坐标和选取的系数都相同,模加运算的结果相同。映射算法过程产生的碰撞不予考虑。这个过程碰撞是由 hash 运算产生的,而2007年的欧洲密码年会 CPK 的审阅报告认为:hash 的安全问题不应作为 CPK 的安全问题。针对模加运算的结果相同,文献[12,13]都设计了无碰撞的私钥矩阵构造方法,他们限制了基点群阶数 n 的大小,使得每个私钥都满足 $(\sum_{j=1}^h r_{ij,j}) \bmod n = \sum_{j=1}^h r_{ij,j}$,这与本文提出的解决共谋攻击的 CPK 算法相抵触,而且他们的方案无法避免密钥更新后的密钥碰撞问题。因此,本文提出新的解决密钥碰撞的方法。



图1 私钥的生成过程

4.2 解决密钥碰撞的方案

密钥碰撞即密钥相同,私钥相同,公钥也相同。私钥都由每个用户单独保管,无法对其进行比较,但是用户的公钥是可以公开的,可以进行比较。那么,在生成私钥之前,先生成公钥,然后与已注册用户的公钥对比。若存在与已注册用户相同的公钥,则要求用户更改所申请的身份标志。密钥生成的流程如图2所示。

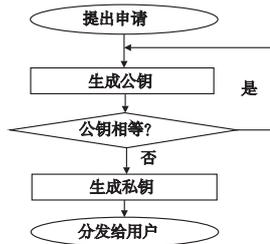


图2 密钥的生成流程

用户的身份标志有时可能是不能更改的,如身份证号码、手机号码、IP 地址等,此时就不能通过更改用户所申请的身份标志。此时采用下面的算法来避免密钥碰撞:

- 选择随机数 r , 计算 $sk' = (sk + r) \bmod n_0$ 。
- 计算公钥增量 $\Delta PK = rG$ 。
- 计算公钥 $PK' = PK + \Delta PK$, 与已注册用户公钥比较, 相同则返回 a)。
- 用 sk 对公钥增量签名 $\text{sign}_{sk}(\Delta PK)$, 公开 ΔPK 和 $x\text{sign}_{sk}(\Delta PK)$ 。

本算法可以解决所有的密钥碰撞问题,即便是用户的标志相同。不管密钥生成过程中是否产生了相同的中间量,都可以通过公钥的对比得知密钥的碰撞情况,进而添加随机数来改变私钥,直到产生出与已注册用户不同的私钥。

5 算法分析

本算法为解决线性共谋攻击和选择共谋攻击提供了新思路,并提升了抵御随机共谋攻击性能,同时解决了密钥碰撞问题。与原 CPK 算法相比,存储量、计算量、通信量都有一定程度的增加。

5.1 存储量

设公私钥矩阵的规模为 $m \times h$, 密钥长度为 256 bit (32 Byte), 系数序列 (a_1, a_2, \dots, a_t) 长度为 t 的 $k(2^k > t)$ 比特整数序列。用户的 ID 证书中存储了私钥及个人相关信息,如证书名、授权等级、所属单位等,与原 CPK 算法相比, ID 证书内容不变,单个用户存储量不变。公钥公开的是计算公钥的变量,而不是直接公开公钥。公钥变量包括公钥矩阵、系数序列、hash 函数、映射函数等,存储量很小。公钥变量公开在最容易访问的媒体上,如读卡设备(ATM 机、POS 机)、移动终端、网站,甚至写在芯片上直接分发给用户。密钥管理中心要存储用户的公钥,用于在生产密钥时检测是否与已注册用户产生了碰撞。假设系统用户为 1 000 万,那么公钥存储量为 $2 \times 32 \times 10000000 \text{ Byte} \approx 640 \text{ MB}$ 。对于现今的密钥管理中心而言,此公钥存储量可以接受。

5.2 计算量

本算法在一定程度上增加了计算量。首先与原 CPK 算法相比,在密钥产生的流程上增加了对比公钥的过程;其次,密钥产生的过程中增加了一次 hash 运算和 h 次模乘运算。私钥的产生是在密钥管理中心进行的,运算量的增加对密钥管理中心影响极小。实际情况中,终端设备需要计算公钥,运算量增加造成的公钥产生的延时对系统影响很小。

5.3 通信量

在生产密钥的过程中若与已注册用户存在密钥碰撞,则会选择随机数重新生产的密钥,从而此密钥具有一定的随机性,与用户的身份标志没有绑定关系。用户间通信时需要先对公钥增量的签名进行验证,增加了通信量。只有存在密钥碰撞时才会使通信量增加,由文献[10]可知,密钥碰撞的概率是极小的。同时,也只有与产生了密钥碰撞的用户进行通信时才会导致通信量的增加,对其他用户的通信量没有影响。

本文的方案对密钥管理中心而言,在存储量、计算量上有所增加,但增加量都不大,且可以接受,不存在通信量。对终端设备而言,存储量、计算量、通信量略有增加,增加量都很小,可以接受。对用户而言,计算和用户间的通信不在用户的证书内而是在终端设备上,因此不存在计算量和通信量。而证书的存储量内容没有变化,存储量不变。总体而言,本方案存储量、计算量、通信量的增加对系统的影响很小。

6 结束语

本算法通过构造种子矩阵,使种子密钥和大于 n , 消除了密钥间的线性关系,从而使密钥间不能相互线性表示,解决了线性共谋攻击问题。在密钥生产过程中,通过引入系数破坏了层不同和层互斥不同的关系,解决了选择共谋攻击问题,同时一定程度上提升了抵抗随机共谋攻击的性能。针对密钥碰撞问题提出了公钥对比算法,解决了密钥碰撞问题。但本算法仍存在随机共谋攻击,并且增加了一定的存储量、计算量、通信量。解决共谋攻击问题和提高密钥产生的效率仍值得进一步研究。

参考文献:

- [1] NAN Xiang-hao. CPK cryptosystem and identity authentication [M]. Beijing: Publishing House of Electronic Industry, 2012.
- [2] NAN Xiang-hao. Identity authentication, technical basis of cyber security [M]. Beijing: Publishing House of Electronics and Industry, 2011.

一次同时对 l 个 bit 位进行加解密和 DGHV 方案一次对 1 个 bit 位进行加解密的代价是一样的,因此本文方案具有加解密效率更高、密文扩展率低和公/私钥尺寸短等特点,更易于在实际的密码系统中应用。相对于 BDGHV 方案,本文方案的公/私钥尺寸更短。

5 结束语

到目前为止,虽然全同态加密方案还不能在实际的密码系统中应用,但自 Gentry 构造出第一个基于理想格的全同态加密方案后,全同态加密已成为当前研究的热点,而研究的方向主要集中在提高方案的效率和安全性这两个方面。在效率方面的研究主要是提高方案的加解密效率和压缩方案的公钥尺寸。本文提出一个具有较短公钥尺寸的批处理整数上的全同态加密方案,并将其语义安全规约到零错误的近似最大公因子问题。相对于 Dijk 等人^[9]的整数全同态方案,本方案具有较短的公钥尺寸、较高的加解密效率;相对于 Coron 等人^[13]的批处理整数上的全同态加密方案,本方案具有较短的公钥尺寸。

参考文献:

- [1] RIVEST R L, ADLEMAN L, DERTOUZOS M L. On data banks and privacy homomorphisms[C]//Proc of the 17th IEEE Annual Symposium on Foundations of Computer Science. [S. l.]:Academic Press, 1978:169-177.
- [2] GENTRY C. Fully homomorphic encryption using ideal lattices[C]//Proc of the 41st ACM Symposium on Theory of Computing. New York: ACM Press, 2009:169-178.
- [3] GENTRY C. A fully homomorphic encryption scheme[D]. Stanford: Stanford University, 2009.
- [4] SMART N P, VERCAUTEREN F. Fully homomorphic encryption with relatively small key and ciphertext sizes[C]//Proc of the 13th International Conference on Practice and Theory in Public Key Cryptography. 2010:420-443.
- [5] STEHLÉ D, STEINFELD R. Faster fully homomorphic encryption [C]//Proc of ASICRYPT. 2010:377-394.
- [6] BRAKERSKI Z. Fully homomorphic encryption without modulus switching from classical GapSVP [C]//Advances in Cryptology-CRYPTO. Berlin:Springer, 2012:868-886.
- [7] SMART N P, VERCAUTEREN F. Fully homomorphic SIMD operations[C]//Designs, Codes and Cryptography. [S. l.]: Springer, 2012:1-25.
- [8] GENTRY C, HALEVI S. Implementing Gentry's fully-homomorphic encryption scheme [C]//Proc of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology. Berlin:Springer-Verlag, 2011:129-148.
- [9] VAN DIJK M, GENTRY C, HALEVI S, et al. Fully homomorphic encryption over the integers [C]//Advances in Cryptology-EUROCRYPT. Berlin:Springer, 2010:24-43.
- [10] CORON J S, MANDAL A, NACCACHE D, et al. Fully homomorphic encryption over the integers with shorter public keys [C]//Advances in Cryptology-CRYPTO. Berlin:Springer, 2011:487-504.
- [11] CORON J S, NACCACHE D, TIBOUCHI M. Optimization of fully homomorphic encryption [C]//IACR Cryptology ePrint Archive. 2011:440.
- [12] CORON J S, NACCACHE D, TIBOUCHI M. Public key compression and modulus switching for fully homomorphic encryption over the integers [C]//Advances in Cryptology-EUROCRYPT. Berlin: Springer, 2012:446-464.
- [13] CORON J S, LEPOINT T, TIBOUCHI M. Batch fully homomorphic encryption over the integers [C]//IACR Cryptology ePrint Archive. 2013.
- [14] BRAKERSKI Z, VAIKUNTANATHAN V. Efficient fully homomorphic encryption from (standard) LWE [C]//Proc of the 52nd IEEE Annual Symposium on Foundations of Computer Science. 2011:97-106.
- [15] BRAKERSKI Z, VAIKUNTANATHAN V. Fully homomorphic encryption from ring-LWE and security for key dependent messages [C]//Advances in Cryptology-CRYPTO. Berlin:Springer, 2011:505-524.
- [16] GENTRY C. Fully homomorphic encryption without bootstrapping[J]. Security, 2011, 111(111):1-12.
- [17] GENTRY C, HALEVI S, SMART N P. Fully homomorphic encryption with polylog overhead [C]//Proc of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques. Berlin:Springer-Verlag, 2012:465-482.
- [18] GENTRY C, HALEVI S, SMART N. Better bootstrapping in fully homomorphic encryption [C]//Proc of the 15th International Conference on Practice and Theory in Public Key Cryptography. 2012:1-16.
- [19] 光焱, 顾纯祥, 祝跃飞, 等. 一种基于 LWE 问题的无证书全同态加密体制[J]. 电子与信息学报, 2013, 35(4):988-993.
- [20] CHEN Yuan-mi, NGUYEN P Q. Faster algorithms for approximate common divisors: breaking fully-homomorphic-encryption challenges over the integers [C]//Proc of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques. Berlin:Springer, 2012:502-519.
- [21] 吴晓园. 基于格的全同态加密方案的研究与设计[D]. 西安:西安电子科技大学, 2012.
- [22] 赵美玲, 张少武. 基于 ECC 的组合公钥技术的安全性分析[J]. 计算机工程, 2008, 34(1):156-157.
- [23] 邵春雨, 苏锦海, 魏有国, 等. 一种双矩阵组合公钥算法[J]. 电子学报, 2011, 39(3):671-674.
- [24] 邵春雨. 双矩阵组合公钥算法及应用研究[D]. 郑州:解放军信息工程大学, 2010.
- [25] 王公浩, 王玟, 吴铎, 等. CPK 随机碰撞概率分析[J]. 信息安全与通信保密, 2008(11):87-88.
- [26] 荣昆, 李益发. CPK 种子矩阵的优化设计方案[J]. 计算机工程与应用, 2006, 42(24):120-121.
- [27] 邢海龙. 组合公钥 CPK 关键技术研究与应用[D]. 长沙:国防科技大学, 2009.
- [28] 马芯宇, 龙翔, 范修斌. 无碰撞 CPK 的种子库构建和选取方案[J]. 计算机工程与应用, 2012, 48(27):99-104.

(上接第 1179 页)

- [3] 南湘浩. CPK 组合公钥体制 (v8.0) [J]. 信息安全与通信保密, 2013(3):39-41.
- [4] 南湘浩. CPK 算法与标识认证 [J]. 信息安全与通信保密, 2006(9):12-16.
- [5] 张福泰, 孙银霞, 张磊, 等. 无证书公钥密码体制研究 [J]. 软件学报, 2011, 22(6):1316-1332.
- [6] 侯孟波. 基于身份和无证书的两方认证密钥协商协议研究 [D]. 济南:山东大学, 2010.
- [7] 胡亮, 初剑峰, 林海群, 等. IBE 体系的密钥管理机制 [J]. 计算机学报, 2009, 4(3):544-556.
- [8] 南湘浩. CPK 标识认证 [M]. 北京:国防工业出版社, 2006.
- [9] 赵建国. 组合公钥 (CPK) 技术的创新实践 [J]. 信息安全与通信保密, 2012(5):55-57.