$$col1_i = col1_{i-1},$$
$$sum_i = col1_i + \sum_{j=r+1}^{i} \beta_{ji}^{(f)} ver_j (i = r+1,...,r+g).$$

将第 r 列到第 2 列分别与扩展向量 ver 相乘,有:

$$col1_i = col1_{i-1} + \alpha_{ii}^{(f)} ver_i,$$
$$sum_i = col1_i (i = 2,...,r).$$

最后,对于第 1 列的计算有:

$$col1_1 = \alpha_{11}^{(f)} ver_1,$$
$$sum_1 = col1_1.$$

将上述步骤中计算结果 $sum_i(i=1,...,n+1)$与扩展向量 ver 的转置相乘,得到消息值、签名值代入第 f 个公钥方程的计算结果:

$$h_f = ver \cdot PM[f] \cdot ver^T = \sum_{j=1}^{n+1} sum_j ver_j.$$

3. 最后,验证这 g 个计算结果:

$$h_f=0, \forall f \in \{1,\ldots,g\}.$$

如果上式成立,则签名有效;否则,签名无效.

附录 B 的算法 B1 给出签名验证的伪代码,对应上述验证流程.

2.3 CyclicRGB混合多变量签名方案

在具体分析了 CyclicRGB 的签名验证之后,我们就可以设计 CyclicRGB 混合多变量签名方案.CyclicRGB 混合多变量签名方案由 3 种多项式时间算法组成,分别为密钥生成算法、签名生成算法和签名验证算法.

密钥生成算法. (Key Generation):$(pk,sk) \leftarrow KeyGeneration(1^{\lambda})$.

输入:安全参数λ.

输出:用户的公钥 pk、私钥 sk.

RGB 公钥方程组中的每个方程的系数矩阵以及中心映射 F 的每个方程的系数矩阵都可以表示为$(n+1)\times(n+1)$的上三角矩阵形式,均为第 2.2 节中 $PM[k]$矩阵表示形式.不同的是,在中心映射 F 的每一个方程的系数矩阵中,表示 *Green* 变量与 *Green* 变量乘积的二次项系数全部为 0;相反地,对应的公钥方程的系数矩阵中,表示 *Green* 变量与 *Green* 变量乘积的二次项系数需要由私钥求得.

在密钥生成过程中,首先生成公钥方程组系数矩阵中的二次项系数部分,利用公钥方程组的系数矩阵与中心映射的系数矩阵的线性关系,求得中心映射系数矩阵的二次项系数.当中心映射系数矩阵的二次项系数确定以后,再为中心映射 F 随机选取一次项系数和常数项系数.最后,由中心映射 F 和可逆仿射变换 S_0,S_3 确定最终公钥方程.具体密钥生成过程如下.

1. 随机选择两个向量 $v \in K^{r(n+g+b+1)/2}, w \in K^{b(b+1)/2}$.随机选择 3 个可逆仿射变换 $S_1:K^r \to K^r, S_2:K^{g+b} \to K^{g+b}, S_3:K^g \to K^g$.通过第 1.2.2 节中的方法,由 S_1,S_2 求出 S_0.并使用附录 A 中的方法,由 S_0 求得矩阵 A.
2. 使用 v,w,按照公式(3)循环右移生成每个公钥方程系数矩阵中对应的二次项系数.即生成公钥系数矩阵 $Pm=(V|U|W|C)$中的矩阵 V 和 W.
3. 公钥系数矩阵中除了 V 和 W 表示二次项系数之外,U 也表示二次项系数,并且 U 表示 *Green* 变量分别与 *Green* 变量、*Blue* 变量构成的二次项乘积的系数矩阵.对于这一部分系数,我们将 U 中表示 *Green* 变量与 *Green* 变量组成的二次项的系数全部设置为 0.公钥系数矩阵中的这一部分值最终由中心映射的一次项系数和常数项系数确定以后求得.
4. 另外,为 U 中表示 *Green* 变量与 *Blue* 变量组成的二次项的系数随机选取值.
5. 通过上述步骤,可以确定公钥方程组中二次项的系数,即 V,U,W.根据公钥方程组的系数矩阵与中心映射系数矩阵的线性关系,通过公式(2)可以得到中心映射 F 的二次项系数.
6. 随机选择中心映射 F 的一次项系数以及常数项系数,根据 $P=S_3 \circ F \circ S_0$ 求得公钥方程的系数矩阵 $Pm=$

$(V|U|W|C)$.CyclicRGB 方案的公钥为 $pk=(v,U,w,C)$,私钥 $sk=(S_1,S_2,S_3,F)$.

7. 输出密钥对(pk,sk).

签名生成算法. (Signature Generation):$X\leftarrow Sign(S_1,S_2,F,M)$.

该算法由签名者执行,签名者根据自己的私钥(S_1,S_2,F)、消息值 $M=(x_1,...,x_r)$.生成相应的消息签名 $X\in K^{g+b}$.具体签名过程如下.

1. 计算 $\tilde{M}=S_1(M)=S_1(x_1,...,x_r)=(x'_1,...,x'_r)$.

2. 随机选择 Blue 变量 $(x'_{r+g+1},...,x'_n)\in K^b$,将这些随机选择的 Blue 变量值与 $\tilde{M}$ 一同代入中心映射 F.使用高斯消元求解未知量 $(x'_{r+1},...,x'_{r+g})\in K^g$ 的值,具有如下形式:

$$\begin{cases} f^{(1)}(x'_1,...,x'_r,x'_{r+1},...,x'_{r+g},x'_{r+g+1},...,x'_n)=0 \\ \vdots \\ f^{(g)}(x'_1,...,x'_r,x'_{r+1},...,x'_{r+g},x'_{r+g+1},...,x'_n)=0 \end{cases}.$$

如果上述线性方程组无解,则重复步骤(2)中的运算过程,即重新选择 *Blue* 变量的值进行运算,直到可以求出 g 个变量 $(x'_{r+1},...,x'_{r+g})$ 的值为止.

3. 将步骤 2 中随机选择的 *Blue* 变量 $(x'_{r+g+1},...,x'_n)\in K^b$ 和步骤 2 中求得的 g 个变量 $(x'_{r+1},...,x'_{r+g})$ 代入 S_2 的逆中,

$$X=S_2^{-1}(X')=S_2^{-1}(x'_{r+1},...,x'_{r+g},x'_{r+g+1},...,x'_n)=(x_{r+1},...,x_{r+g},x_{r+g+1},...,x_n).$$

最后,运算结果 X 即对消息 M 的签名.

签名验证算法. (Signature Verification):{ACCEPT,⊥}←*Verify*(*pk*,*M*,*X*).

该算法由验证者执行,来验证消息签名是否有效.

输入:验证者的公钥 *pk*,消息值 M 以及消息的签名 X.

输出:如果签名有效,返回 ACCEPT;否则签名无效,返回⊥.

为了验证 $X=(x_{r+1},...,x_{r+g},x_{r+g+1},\ldots,x_n)\in F^{g+b}$ 是否为 $M=(x_1,...,x_r)\in F^r$ 的签名,使用第 2.2 节中的验证算法(即算法 B1)对消息和签名进行验证:

$$pk(M,X)=pk(x_1,\ldots,x_r,x_{r+1},...,x_{r+g},x_{r+g+1},\ldots,x_n)=0.$$

若采用算法 B1 的验证结果为“ACCEPT”,则消息签名合法,算法返回 1;否则,消息的签名不合法,算法返回⊥.

2.4 方案正确性证明

CyclicRGB 混合签名方案的正确性可以通过下式加以证明.

$$\begin{aligned} pk(M,X)=0 &\Leftrightarrow S_3\circ F\circ S_0(M,X)=0 \\ &\Leftrightarrow F(S_1(M),S_2(X))=0 \\ &\Leftrightarrow F(\tilde{M},S_2\circ S_2^{-1}(X'))=0 \\ &\Leftrightarrow F(\tilde{M},X')=0 \\ &\Leftrightarrow F(x'_1,...,x'_r,x'_{r+1},...,x'_{r+g},x'_{r+g+1},...,x'_n)=0. \end{aligned}$$

如果消息值和签名值合法,上式一定成立.

3 安全性分析及参数的选择

目前,针对多变量公钥密码方案存在大量的攻击方法.这些攻击方法包括有强力搜索、直接攻击、最小秩攻击、高秩攻击、Patarin 线性关系(线性化方程攻击)、差分攻击、分离“油”“醋”变量攻击、Rainbow Band Seperation 攻击等.下面就 CyclicRGB 方案的安全性进行分析.

由于原始的 RGB 方案是 UOV 方案的改进,本文方案 CyclicRGB 基于原始的 RGB 方案进行改进,在降低

RGB 方案的公钥大小的同时,提高签名验证效率.原始 RGB 方案与本文 CyclicRGB 方案都不是多层次结构的方案,这就使得作用于 Rainbow,CyclicRainbow 等多层次复杂结构方案的攻击方法不能用来攻击 RGB,CyclicRGB 方案.这些攻击方法包括秩攻击、Rainbow Band Separation 等攻击.

本文主要采用循环公钥的方法来降低 RGB 多变量混合签名方案的公钥量.与原方案 RGB 相比,改进后的方案并没有降低原方案的安全性,因此,不能作用于 RGB 方案的攻击方法也不能作用于 CyclicRGB 方案.同时,新方案的安全性也和 RGB 方案一样,可以通过选择合适的参数来抵抗代数攻击.

3.1 穷举攻击

穷举攻击可以作用于任何的密码方案.穷举攻击的方式有两种:一种是穷举密钥空间,通过找到一个等价密钥确定该方法是否可行,这种攻击方法的时间复杂度较高;另一种是对明文直接攻击,通过找到有效的明文的概率来确定这种穷举攻击方法是否可行.明文长度大于 64 比特的方案可以抵抗对明文的穷举攻击.因此,当有限域选择 $GF(2^8)$时,在 CyclicRGB 中表示 *Red* 变量的个数 r,当 $r\geqslant 8$ 时,方案可以抵抗穷举攻击.

3.2 分离油、醋变量的攻击

分离油醋攻击是 Kipnis 等人[19]在 1998 年对 OV 体制进行攻击时提出的攻击方法.文献[6]中指出:当醋变量个数多余油变量个数或者两者个数近似相等时,油醋分离攻击方法的时间复杂度为 $q^{(v-o)-1}\cdot o^4$(其中,v,o 分别为 OV 体制中醋变量的个数和油变量的个数).RGB,CyclicRGB 多变量混合签名方案实质上是对 UOV 签名方案的改进,前两者的中心映射结构与 UOV 方案相同.当将 RGB,CyclicRGB 方案中心映射中的 *Red* 变量和 *Blue* 变量看作醋变量、*Green* 变量看作油变量时,分离油醋攻击方法对于 CyclicRGB 方案的攻击复杂度为 $q^{r+b-g-1}\cdot o^4$.当有限域选择 $GF(2^8)$时,只要 $r+g-b\geqslant 14$,CyclicRGB 的安全级别就将大于 2^{100}.

3.3 Patarin线性关系攻击

Patarin 线性关系攻击最初是针对 MI 多变量公钥密码方案提出的,Patarin 线性攻击方法是对公钥多项式方程进行等价变形,试图使用足够多的明文和密文对得到关于明文变量和密文变量(或者公钥多项式)间的线性关系,最后,攻击者利用这个线性关系达到攻破方案的目的.但是文献[18]中指出,RGB 方案的中心映射不是双射的.因此,线性化方程的攻击方法不使用 RGB;同理,Patarin 线性攻击也不适用于 CyclicRGB.

4 CyclicRGB 与 RGB 公钥大小及签名验证效率的比较

RGB 混合多变量签名方案的公钥大小为

$$g\frac{(n+1)(n+2)}{2}=g\times\frac{r(n+g+b+1)}{2}+g\times\frac{g(2b+g+1)}{2}+g\times\frac{b(b+1)}{2}+g\times(n+1) \tag{4}$$

CyclicRGB 混合多变量签名方案的公钥由向量 v,w 和矩阵 U,C 构成,因此,CyclicRGB 混合多变量签名方案的公钥大小为

$$\frac{r(n+g+b+1)}{2}+\frac{b(b+1)}{2}+g\left[\frac{g(2b+g+1)}{2}+n+1\right]=\frac{r(n+g+b+1)}{2}+g\times\frac{g(2b+g+1)}{2}+\frac{b(b+1)}{2}+g\times(n+1) \tag{5}$$

用公式(4)减去公式(5)得到 $(g-1)\times\frac{r(n+g+b+1)+b(b+1)}{2}$.因此,RGB 的公钥大小比 CyclicRGB 的公钥大 $(g-1)\times\frac{r(n+g+b+1)+b(b+1)}{2}$ 个元素,即 CyclicRGB 方案的公钥大小小于原始方案的公钥大小.

文献[18]对 RGB 方案各部分的时间复杂度进行了分析(包括公钥、私钥的生成,签名的生成和验证),其中包括有限域上的模加法和模乘法运算.但是,有限域上模乘运算的时间复杂度高于有限域上模加法运算的时间复杂度,因此,下面我们主要分析算法中有限域上模乘运算的复杂度.CyclicRGB 验证签名效率的提高主要通过第 2.2 节中签名验证算法得以体现,附录 B 中算法 B1 即为 CyclicRGB 签名验证的伪代码.下面我们就附录 B 中算法 B1 伪代码中用到的模乘运算的个数进行分析.

- 算法 B1 第 1 行~第 3 行需要执行的模乘运算为 $\frac{r(r+1)}{2}+r(g+b-1)$;
- 第 4 行~第 6 行需要执行的模乘运算为 $\frac{b(b-1)}{2}$;
- 第 11 行执行完第 7 行的循环体需要执行的模乘运算数为 $\frac{g(g+1)}{2}$;
- 第 13 行执行完第 7 行的循环体需要执行的模乘运算数为 $g(b-1)$;
- 第 15 行需要执行的模乘运算数为 n;
- 第 16 行需要执行的模乘运算数为 n;
- 第 17 行需要执行的模乘运算数为 $n+1$;
- 从算法 B1 的第 18 行开始的循环语句:
  - 第 19 行需要执行的模乘运算数为 n;
  - 第 20 行需要执行的模乘运算数为 $g+1$;
  - 执行第 21 行的 for 循环中,第 23 行需要执行的模乘运算数为$(b-2)$;
  - 执行第 21 行的 for 循环中,第 24 行需要执行的模乘运算数为 $g(b-2)$;
  - 第 27 行需要执行的模乘运算数为 1;
  - 第 28 行需要执行的模乘运算数为 g;
  - 执行第 29 行的 for 循环中,第 31 行总共执行的模乘运算数为 $\frac{g(g+1)}{2}$;
  - 执行第 33 行的 for 循环中,第 34 行总共执行的模乘运算数为 $r-1$;
  - 第 37 行需要执行的模乘运算数为 1;
  - 第 39 行需要执行的模乘运算数为 $n+1$.

因此,CyclicRGB 混合多变量签名方案的签名验证总共需要执行模乘运算为

$$\frac{n(n+5)}{2}+1+(g-1)\left[\frac{(2b+g)(g+1)}{2}+r+2n+1\right];$$

RGB 混合签名方案在签名验证时所需要执行的模乘运算数为

$$\frac{n(n+1)}{2}+g\times\frac{(n+1)(n+2)-2}{2}.$$

如果有限域 K 选取 $GF(2^8)$,并且选择参数$(r,g,b)=(20,24,10)$,CyclicRGB 混合多变量签名方案在签名验证时所需要的模乘运算仅为 RGB 签名方案模乘运算的 45%.可以看出:CyclicRGB 方案比 RGB 方案签名验证时需要执行的模乘运算数少,验证签名时效率较高.

5 实 验

为了验证 CyclicRGB 签名方案的效率,我们使用 C++对 RGB 签名方案和 CyclicRGB 方案在两组参数情况下进行实现.实验结果见表 1.

实验对两种方案公钥大小、签名验证所消耗时间以及签名验证进行的有限域上的主要模运算时间进行统计.需要说明的是,其中,CyclicRGB 签名验证时间为两部分:括号中的时间主要为签密验证过程中有限域上模运算的总时间;括号外的时间为整个签名验证的时间,这个时间不仅包括签名验证过程中有限域上模运算的时间,还包括使用循环公钥恢复公钥方程对应系数矩阵的时间.在参数相同的情况下,RGB 与 CyclicRGB 所采用的消息、消息的长度以及得到的签名长度都是相同的.例如,在参数取有限域 $GF(2^8)$,$r=20$,$g=24$,$b=10$ 的实验中,消息的大小均为 20B,对消息的签名的大小也均为 34B.不同的是,在参数相同的情况下,我们可以看到,CyclicRGB 签名方案的验证效率高于 RGB 签名方案的验证效率.CyclicRGB 方案与 RGB 方案相比,采用部分循环公钥的方法来设计公钥时,CyclicRGB 方案在签名验证时所需时间约为 RGB 方案签名验证时间的 60%,本文方案可以达到

提高后者签名验证效率的目的.同时,CyclicRGB 方案的公钥大小不超过 RGB 方案公钥大小的 40%,达到了降低公钥大小的目的.

Table 1 Performance comparison of RGB and CyclicRGB

表 1 RGB 方案与 CyclicRGB 方案实现性能比较

方案	RGB(256,20,24,10)	CyclicRGB(256,20,24,10)	RGB(256,28,28,28)	CyclicRGB(256,28,28,28)
公钥大小(KB)	36.09	14.87	99.94	37.19
私钥大小(KB)	31.20	31.20	93.52	93.52
消息长度(B)	20	20	28	28
签名长度(B)	34	34	56	56
密钥生成(s)	10.990 6	10.878 1	46.571 9	46.107 8
签名时间(s)	0.137 6	0.135 9	0.359	0.375
签名验证时间(s)	0.077 8	0.047(0.031 6)	0.219	0.109(0.075 1)

6 结 论

在本文中,我们采用循环公钥的方法对 RGB 方案进行改进,设计了新的混合多变量签名方案——CyclicRGB 方案,并且具体设计了 CyclicRGB 签名验证算法.新的混合多变量签名方案在签名验证时所需要的时间为原始的 RGB 方案签名验证消耗时间的 60%,新方案的公钥大小也仅为 RGB 方案的 40%.我们也采用 C++对有限域 $GF(2^8)$上不同参数情况下的两种方案的签名效率进行了实验比较.实验结果表明:CyclicRGB 的签名方案在验证签名时,可以在很大程度上提高签名验证的效率.

References:

[1] Shor PW. Polynomial-Time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 1999,41(2):303−332. [doi: 10.1137/S0036144598347011]

[2] Matsumoto T, Imai H. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Günther CG, ed. Proc. of the Workshop on the Theory and Application of of Cryptographic Techniques. Davos: Springer-Verlag, 1988. 419−453. [doi:10.1007/3-540-45961-8_39]

[3] Partarin J. Hidden field equations (HFE) and isomorphism of polynomial (IP): Two new families of asymmetric algorithms. In: Maurer U, ed. Proc. of the Advances in Cryptology—EUROCRYPT'96. LNCS 1070, Berlin, Heidelberg: Springer-Verlag, 1996. 33−48. [doi: 10.1007/3-540-68339-9_4]

[4] Kipnis A, Shamir A. Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener M, ed. Proc. of the Advances in Cryptology—CRYPTO'99. Springer-Verlag, 1999. 19−30. [doi:10.1007/3-540-48405-1_2]

[5] Ding J, Gower JE, Schmidt DS. Oil-Vinegar signature schemes. In: Ding J, Gower JE, Schmidt DS, eds. Proc. of the Multivariate Public Key Cryptosystems, Vol.25. Springer-Verlag, 2006. 63−97. [doi: 10.1007/978-0-387-36946-4_3]

[6] Kipnis A, Patarin J, Goubin L. Unbalanced oil and vinegar signature schemes. In: Stern J, ed. Proc. of the Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Prague: Springer-Verlag, 1999. 206−222. [doi: 10.1007/3-540-48910-X_15]

[7] Patarin J, Courtois N, Goubin L. Flash, a fast multivariate signature algorithm. In: Naccache D, ed. Proc. of the Cryptographers' Track at the RSA Conf. San Francisco: Springer-Verlag. 2001. 298−307. [doi: 10.1007/3-540-45353-9_22]

[8] Ding J, Schmidt D. Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis J, ed. Proc. of the Int'l Conf. on Applied Cryptography and Network Security. New York: Springer-Verlag, 2005. 164−175. [doi: 10.1007/11496137_12]

[9] Ding J. A new variant of the Matsumoto-Imai cryptosystem through perturbation. In: Bao F, ed. Proc. of the Int'l Workshop on Public Key Cryptography. Singapore: Springer-Verlag, 2004. 305−318. [doi: 10.1007/978-3-540-24632-9_22]

[10] Ding J, Gower JE. Inoculating multivariate schemes against differential attacks. In: Yung M, ed. Proc. of the Int'l Workshop on Public Key Cryptography. New York: Springer-Verlag, 2006. 290−301. [doi: 10.1007/11745853_19]

[11] Porras J, Baena J, Ding J. ZHFE, a new multivariate public key encryption scheme. In: Mosca M, ed. Proc. of the Int'l Workshop on Post-Quantum Cryptography. Waterloo: Springer Int'l Publishing, 2014. 229−245. [doi: 10.1007/978-3-319-11659-4_14]

[12] Tao C, Diene A, Tang S, Ding J. Simple matrix scheme for encryption. Lecture Notes in Computer Science, 2013,7932:231−242.

[13] Ding J, Petzoldt A, Wang L. The cubic simple matrix encryption scheme. In: Mosca M, ed. Proc. of the Int'l Workshop on Post-Quantum Cryptography. Waterloo: Springer Int'l Publishing, 2014. 76–87. [doi: 10.1007/978-3-319-11659-4_5]

[14] Yasuda T, Sakurai K. A multivariate encryption scheme with rainbow. In: Qing SH, ed. Proc. of the Int'l Conf. on Information and Communications Security. Beijing: Springer Int'l Publishing, 2015. 236–251. [doi: 10.1007/978-3-319-29814-6_19]

[15] Petzoldt A, Bulygin S, Buchmann J. CyclicRainbow—A multivariate signature scheme with a partially cyclic public key. In: Gong G, ed. Proc. of the Int'l Conf. on Cryptology in India. Hyderabad: Springer-Verlag, 2010. 33–48. [doi: 10.1007/978-3-642-17401-8_4]

[16] Petzoldt A, Bulygin S, Buchmann J. Fast verification for improved versions of the UOV and rainbow signature schemes. In: Gaborit P, ed. Proc. of the Int'l Workshop on Post-Quantum Cryptography. Limoges: Springer-Verlag, 2013. 188–202. [doi: 10.1007/978-3-642-38616-9_13]

[17] Duong DH, Petzoldt A, Takagi T. Reducing the key size of the SRP encryption scheme. In: Liu JK, ed. Proc. of the Australasian Conf. on Information Security and Privacy. Melbourne: Springer Int'l Publishing, 2016. 427–434. [doi: 10.1007/978-3-319-40367-0_27]

[18] Shen W, Tang S. RGB, a mixed multivariate signature scheme. The Computer Journal, 2016,59(4):439–451. [doi: 10.1093/comjnl/bxv056]

[19] Kipnis A, Shamir A. Cryptanalysis of the oil and vinegar signature scheme. In: Krawczyk H, ed. Proc. of the Annual Int'l Cryptology Conf. Springer-Verlag, 1998. 257–266. [doi: 10.1007/BFb0055733]

附录 A

下面证明:在已知多变量二次多项式方程 Q 和可逆仿射变换 S_0 的情况下,Q 和 F 的系数矩阵之间存在线性关系.由本文第 2 节可知,

$$Qm = F \circ S_0 = \begin{cases} S_0^T \times FM[1] \times S_0 \\ \vdots \\ S_0^T \times FM[g] \times S_0 \end{cases}, \ Q = F \circ S_0.$$

这里假设 $S_0 = \begin{pmatrix} s_{11}, s_{12}, ..., s_{1n} \\ s_{21}, s_{22}, ..., s_{2n} \\ \vdots \\ s_{n1}, s_{n2}, ..., s_{nn} \end{pmatrix}, FM[k] = \begin{pmatrix} f_{11}^{(k)}, f_{12}^{(k)}, ..., f_{1n}^{(k)} \\ f_{21}^{(k)}, f_{22}^{(k)}, ..., f_{2n}^{(k)} \\ \vdots \\ f_{n1}^{(k)}, f_{n2}^{(k)}, ..., f_{nn}^{(k)} \end{pmatrix}$. 假设 $QM[k]$为 Q 中第 k 个方程的系数矩阵,则 $QM[k] = S_0^T \times FM[k] \times S_0$, 并且

$$QM[k] = \begin{pmatrix} q_{11}^{(k)}, q_{12}^{(k)}, ..., q_{1n}^{(k)} \\ q_{21}^{(k)}, q_{22}^{(k)}, ..., q_{2n}^{(k)} \\ \vdots \\ q_{n1}^{(k)}, q_{n2}^{(k)}, ..., q_{nn}^{(k)} \end{pmatrix}$$

$$= \begin{pmatrix} s_{11}, s_{12}, ..., s_{1n} \\ s_{21}, s_{22}, ..., s_{2n} \\ \vdots \\ s_{n1}, s_{n2}, ..., s_{nn} \end{pmatrix}^T \times \begin{pmatrix} f_{11}^{(k)}, f_{12}^{(k)}, ..., f_{1n}^{(k)} \\ f_{21}^{(k)}, f_{22}^{(k)}, ..., f_{2n}^{(k)} \\ \vdots \\ f_{n1}^{(k)}, f_{n2}^{(k)}, ..., f_{nn}^{(k)} \end{pmatrix} \times \begin{pmatrix} s_{11}, s_{12}, ..., s_{1n} \\ s_{21}, s_{22}, ..., s_{2n} \\ \vdots \\ s_{n1}, s_{n2}, ..., s_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} s_{11}, s_{21}, ..., s_{n1} \\ s_{12}, s_{22}, ..., s_{n2} \\ \vdots \\ s_{1n}, s_{2n}, ..., s_{nn} \end{pmatrix} \times \begin{pmatrix} f_{11}^{(k)}, f_{12}^{(k)}, ..., f_{1n}^{(k)} \\ f_{21}^{(k)}, f_{22}^{(k)}, ..., f_{2n}^{(k)} \\ \vdots \\ f_{n1}^{(k)}, f_{n2}^{(k)}, ..., f_{nn}^{(k)} \end{pmatrix} \times \begin{pmatrix} s_{11}, s_{12}, ..., s_{1n} \\ s_{21}, s_{22}, ..., s_{2n} \\ \vdots \\ s_{n1}, s_{n2}, ..., s_{nn} \end{pmatrix}.$$

那么, $q_{ij}^{(k)}$ 为矩阵 $S_0^T \times FM[k]$ 乘积结果的第 i 行与 S_0 的第 j 列相乘的结果.S_0 第 j 列可以表示为 $(s_{1j}, s_{2j}, \ldots, s_{nj})^T$,

而矩阵 $S_0^T \times FM[k]$ 乘积结果的第 i 行第 c 列(c=1,...,n)的元素由 S_0^T 的第 i 行元素分别与 $FM[k]$的第 c 列对应元素乘积之和得到.又因为 S_0^T 为 S_0 的转置矩阵,因此, S_0^T 的第 i 行就为 S_0 的第 i 列,即$(s_{1i},s_{2i},\ldots,s_{ni})^T$.可以求得 $q_{ij}^{(k)}$ 为

$$
\begin{aligned}
q_{ij}^{(k)} &= (s_{1i},s_{2i},...,s_{ni})\times\begin{pmatrix} f_{11}^{(k)},f_{12}^{(k)},...,f_{1n}^{(k)} \\ f_{21}^{(k)},f_{22}^{(k)},...,f_{2n}^{(k)} \\ \vdots \\ f_{n1}^{(k)},f_{n2}^{(k)},...,f_{nn}^{(k)} \end{pmatrix}\times\begin{pmatrix} s_{1j} \\ s_{2j} \\ \vdots \\ s_{nj} \end{pmatrix} \\
&= \begin{pmatrix} s_{1i}f_{11}^{(k)}+s_{2i}f_{21}^{(k)}+...+s_{ni}f_{n1}^{(k)} \\ s_{1i}f_{12}^{(k)}+s_{2i}f_{22}^{(k)}+...+s_{ni}f_{n2}^{(k)} \\ \vdots \\ s_{1i}f_{1n}^{(k)}+s_{2i}f_{2n}^{(k)}+...+s_{ni}f_{nn}^{(k)} \end{pmatrix}^T\times\begin{pmatrix} s_{1j} \\ s_{2j} \\ \vdots \\ s_{nj} \end{pmatrix} \\
&= (s_{1i}s_{1j}f_{11}^{(k)}+s_{2i}s_{1j}f_{21}^{(k)}+...+s_{ni}s_{1j}f_{n1}^{(k)})+(s_{1i}s_{2j}f_{12}^{(k)}+s_{2i}s_{2j}f_{22}^{(k)}+...+s_{ni}s_{2j}f_{n2}^{(k)})+...+ \\
&\quad (s_{1i}s_{nj}f_{1n}^{(k)}+s_{2i}s_{nj}f_{2n}^{(k)}+...+s_{ni}s_{nj}f_{nn}^{(k)}) \\
&= \sum_{r=1}^{n}\sum_{c=1}^{n}s_{ri}s_{cj}f_{rc}^{(k)} \\
&= (f_{11}^{(k)},...,f_{1n}^{(k)},f_{21}^{(k)},...,f_{2n}^{(k)},...,f_{n1}^{(k)},...,f_{nn}^{(k)})\times(s_{1i}s_{1j},...,s_{1i}s_{nj},s_{2i}s_{1j},...,s_{2i}s_{nj},...,s_{ni}s_{1j},...,s_{ni}s_{nj})^T.
\end{aligned}
$$

因此,可以得到如下关系:

$$
\begin{aligned}
Qm &= \begin{pmatrix} q_{11}^{(1)}\ q_{12}^{(1)}...\ q_{1n}^{(1)}\ q_{21}^{(1)}...\ q_{2n}^{(1)}...\ q_{n1}^{(1)}...\ q_{nn}^{(1)} \\ \vdots \\ q_{11}^{(k)}\ q_{12}^{(k)}...\ q_{1n}^{(k)}\ q_{21}^{(k)}...\ q_{2n}^{(k)}...\ q_{n1}^{(k)}...\ q_{nn}^{(k)} \end{pmatrix} \\
&= \begin{pmatrix} f_{11}^{(1)}\ f_{12}^{(1)}...\ f_{1n}^{(1)}\ f_{21}^{(1)}...\ f_{2n}^{(1)}...\ f_{n1}^{(1)}...\ f_{nn}^{(1)} \\ \vdots \\ f_{11}^{(k)}\ f_{12}^{(k)}...\ f_{1n}^{(k)}\ f_{21}^{(k)}...\ f_{2n}^{(k)}...\ f_{n1}^{(k)}...\ f_{nn}^{(k)} \end{pmatrix}\times\begin{pmatrix} s_{11}s_{11}\ \ s_{11}s_{12}\cdots s_{11}s_{1n}\ \ s_{12}s_{11}\cdots s_{12}s_{1n}\cdots\ s_{1n}s_{11}\cdots s_{1n}s_{1n} \\ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots \\ s_{11}s_{n1}\ s_{11}s_{n2}\cdots s_{11}s_{nn}\ \ s_{12}s_{n1}\cdots s_{12}s_{nn}\cdots\ s_{1n}s_{n1}\cdots s_{1n}s_{nn} \\ s_{21}s_{11}\ s_{21}s_{12}\cdots s_{21}s_{1n}\ \ s_{22}s_{11}\cdots s_{22}s_{1n}\cdots\ s_{2n}s_{11}\cdots s_{2n}s_{1n} \\ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots \\ s_{21}s_{n1}\ s_{21}s_{n2}\cdots s_{21}s_{nn}\ \ s_{22}s_{n1}\cdots s_{22}s_{nn}\cdots\ s_{2n}s_{n1}\cdots s_{2n}s_{nn} \\ s_{n1}s_{11}\ s_{n1}s_{12}\cdots s_{n1}s_{1n}\ \ s_{n2}s_{11}\cdots s_{n2}s_{1n}\cdots\ s_{nn}s_{11}\cdots s_{nn}s_{1n} \\ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots\ \ \vdots \\ s_{n1}s_{n1}\ s_{n1}s_{n2}\cdots s_{n1}s_{nn}\ \ s_{n2}s_{n1}\cdots s_{n2}s_{nn}\cdots\ s_{nn}s_{n1}\cdots s_{nn}s_{nn} \end{pmatrix} \\
&= Fm\times A.
\end{aligned}
$$

多变量多项式方程组 Q 的系数矩阵和 F 的系数矩阵之间存在线性关系,即 Qm=Fm×A.这里,A 为关于 S_0 的元素的矩阵.

附录 B

- CyclicRGB 签名验证算法

算法 B1. CyclicRGB 签名的验证.

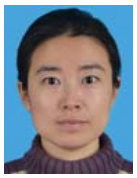
1. for i=1 to n−1 //将扩展向量 ver 代入公钥方程组中第 1 个多项式方程计算
2. $\quad col1_i = \sum_{j=1}^{\min(i,r)}\alpha_{ji}^{(1)}ver_j$
3. end for
4. for i=r+g+1 to n−1
5. $\quad col2_i = \sum_{j=r+g+1}^{i}\gamma_{ji}^{(1)}ver_j$
6. end for

7. for i=1 to n−1
8. if $i \leqslant r$
9. $sum_i=col1_i$
10. else if $i \leqslant r+g$
11. $sum_i = col1_i + \sum_{j=r+1}^{i} \beta_{ji}^{(1)} ver_j$
12. else
13. $sum_i = col1_i + col2_i + \sum_{j=r+1}^{r+g} \beta_{ji}^{(1)} ver_j$
14. end for
15. $sum_n = \sum_{j=1}^{r} \alpha_{jn}^{(1)} ver_j + \sum_{j=r+1}^{r+g} \beta_{jn}^{(1)} ver_j + \sum_{j=r+g+1}^{n} \gamma_{jn}^{(1)} ver_j$
16. $sum_{n+1} = \sum_{j=1}^{n} \eta_j^{(1)} ver_j + \lambda^{(1)}$
17. $h_1 = \sum_{j=1}^{n+1} sum_j ver_j$
18. for f=2 to g //将消息和签名的扩展向量 ver 代入公钥方程组中其余多项式方程计算
19. $sum_{n+1} = \sum_{j=1}^{n} \eta_j^{(f)} ver_j + \lambda^{(f)}$
20. $sum_n = col1_{n-1} + \sum_{j=r+1}^{r+g} \beta_{jn}^{(f)} ver_j + col2_{n-1} + \gamma_{nn}^{(f)} ver_n$
21. for i=n−1 to r+g+2
22. $col1_i=col1_{i-1}$
23. $col2_i = col2_{i-1} + \gamma_{ii}^{(f)} ver_i$
24. $sum_i = col1_i + col2_i + \sum_{j=r+1}^{r+g} \beta_{ji}^{(f)} ver_j$
25. end for
26. $col1_{r+g+1}=col1_{r+g}$
27. $col2_{r+g+1} = \gamma_{(r+g+1)(r+g+1)}^{(f)} ver_{r+g+1}$
28. $sum_{r+g+1} = col1_{r+g+1} + col2_{r+g+1} + \sum_{j=r+1}^{r+g} \beta_{j(r+g+1)}^{(f)} ver_j$
29. for i=r+g to r+1
30. $col1_i=col1_{i-1}$
31. $sum_i = col1_i + \sum_{j=r+1}^{i} \beta_{ji}^{(1)} ver_j$
32. end for
33. for i=r to 2
34. $col1_i = col1_{i-1} + \alpha_{ii}^{(f)} ver_i$
35. $sum_i=col1_i$
36. end for
37. $col1_1 = \alpha_{11}^{f} ver_1$
38. $sum_1=col1_1$
39. $h_f = \sum_{j=1}^{n+1} sum_j ver_j$
40. end for
41. if h_f=0,$\forall f \in \{1,\ldots,g\}$
42. return “ACCEPT”
43. else
44. return “⊥”

45. end if

- 算法 B1 的工作流程

算法 B1 是将消息和签名构成的扩展向量 *ver* 代入循环结构公钥方程组的每个方程中进行计算的过程.其中,第 1 行~第 17 行是将 *ver* 代入公钥方程组中第 1 个方程的运算过程,即 $ver{\cdot}Pm[1]{\cdot}ver^T.sum$ 保存 $ver{\cdot}Pm[1]$的计算结果.同时,由于公钥方程组中方程系数矩阵 $PM[k]$的α,γ部分分别由向量 v,w 循环右移生成,因此在计算的过程中,将这部分的计算结果保存在 $col1_i,col2_i$ 中,用于下一步 $ver{\cdot}Pm[2]$的计算.算法的第 18 行~第 40 行是将 *ver* 代入剩下的 $g-1$ 个方程进行验证的过程.例如,当 $f=2$ 时,对多变量公钥方程组中的第 2 个方程进行计算.在这一步计算过程中,我们利用了上一步计算结果的 $col1_i,col2_i$,这样就可以节省一些模乘运算,提高验证效率.计算过程中,根据现有公钥方程系数矩阵对 $col1_i,col2_i$ 值进行更新,目的是用于下一个多项式方程的计算.CyclicRGB 方案的签名验证就是通过这些值在第 2 个~第 g 个公钥方程计算的过程中的重复利用来降低签名验证的计算代价,从而提高了签名验证效率.对于 $f=2$,算法第 18 行~第 38 行对应 $ver{\cdot}Pm[2]$的计算.第 39 行为公钥方程组中第 2 个方程的计算结果 $ver{\cdot}Pm[2]{\cdot}ver^T$.当 $f=2$ 到 $f=g$ 循环全部结束时,CyclicRGB 验证签名的计算结束.最后,第 41 行~第 45 行判断签名的有效性,并返回判断结果.



李慧贤(1977－),女,内蒙古乌兰浩特人,博士,副教授,主要研究领域为信息安全,多变量公钥密码,安全协议设计与分析.

庞辽军(1978－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为密码学,信息安全.

王凌云(1991－),男,硕士,主要研究领域为多变量公钥密码,安全协议设计与分析.