**CPK组合公钥**

南相浩 陈钟

内容提要： CPK组合公钥体制为现存的公钥体制提供一种使之变为标识密钥体制的通用方法。CPK公钥体制又称CPK标识密钥，只要给定一个标识，就能生成公私钥对，不仅能解决规模化密钥管理难题，也能解决标识鉴别[1]的难题，同时讨论了在数字签名协议和密钥加密协议上防量子计算攻击的途径。

关键字：CPK组合公钥 CPK标识密钥 标识鉴别

在现代公钥体制中，公钥的分发一直是一个难题。1984年，Shamir 第一次提出标识直接作为公钥的问题[2]，并用整数分解难题构建了著名的基于标识的公钥体制IBC。这使公钥体制的发展向前推进了一大步。根据真值逻辑（truth logic）[3][4]，实体分为标识和本体，标识指实体的唯一名称。1989年，中国第一次使用了基于标识的公钥体制，即用RSA构建的组合公钥CPK-RSA体制，应用于大型国防网络中，2003年以“多重公钥”的名义公布[5]。CPK已形成为大家族，包括基于因子分解难题的CPK-RSA, 基于离散对数难题的CPK-DLP, 基于椭圆曲线难题的CPK-ECC, 基于圆锥曲线的CPK-CCC [6], 基于双线对的CPK-BLP等。其中CPK-ECC的签名做的最短。

量子计算机的出现，给公钥体制提出了新的挑战，必须走出依靠计算复杂度的思路。CPK提供了公钥体制中加入非线性运算的可能性，可以将公钥体制建立在线性复杂度的基础上。

下面以CPK-ECC为例，说明其工作原理。

# **1 组合原理**

CPK-ECC是在ECC体制基础上实现的基于标识的非对称公众密钥体制，密钥分为私有密钥和公有密钥。在有限域F*p*上，椭圆曲线E: *y*2 ≡ ( *x*3 + *ax* + *b* ) mod *p*由参数 (*a*, *b*, G, *n*, *p*) 定义。其中*a*, *b*是系数，*a*, *b*, *x*, *y* ∈F*p*，G为加法群的基点，*n*是以G为基点的群的阶。令任意小于*n*的整数为私钥，则*r* G= *R*为对应公钥。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、私钥对。设私钥之和为

( *r*1 + *r*2 + … + *rm*) mod *n* = *r*

则对应公钥之和为

R1 + R2 + … + R*m* = R (点加)

那么，*r*和*R*刚好形成新的公、私钥对。这是因为，

R = R1 + R2 + … + R*m*= *r*1 G + *r*2 G +…+ *rm* G

=（*r*1 + *r*2 +…+ *rm*）G = *r* G

CPK的组合矩阵分为私钥矩阵和公钥矩阵，分别用( *ri,j* )或( R*i,j* )表示，*r*是小于*n*的随机数。私钥矩阵( *ri,j* )由KMC保有，用于私钥的生成，公钥矩阵由私钥矩阵派生 *r*i,j G = ( *xi,j*, *yi,j* ) = R*i,j*

# **2 组合矩阵**

CPK-soft的组合矩阵有三种，密钥长度分别为192、112、64。组合矩阵分为私钥矩阵a和公钥矩阵A，分别用( *ri,j* )或( Ri*,j* )表示，*r*是小于*n*的32x*h*个不全为线性无关的随机整数。

**私钥矩阵**（a） **公钥矩阵**（A）

*r*1.1 *r*2.1 *r*1.*h* R1.1  R2.1 R1.*h*

*r*2.1 *r2.2* *r*2.*h* R2.1  R*2.2* R2.*h*

a = A=

*r32*.1  *r32*.2 *r32.h* R*32*.1 R*32*.2 ~~R~~*32.h* ，

私钥矩阵( *ri,j* )由KMC保有，用于私钥的生成，公钥矩阵(Ri,j)则由各实体（用户或设备）保有，用于公钥的生成。公钥矩阵要加密保护。

# **3 组合密钥**

组合密钥由组合矩阵产生。公私钥的产生过程完全相同，只是所用矩阵不同。矩阵的大小为(32×32)。

标识到矩阵坐标的映射是由YS序列指示的，YS序列是实体标识(ID)在映射密钥Hkey作用下经Hash变换而得的输出。Hash函数均在密钥下工作，设置三个映射密钥，Hkey1用于192作业，Hkey2用于112作业，Hkey3用于64作业。

YS = HashHkeyi (ID) = *w*0.*w*1,*w*2,…,*w*35；

*w*0..*w*35分成4组，分别以*w*00..*w*08; *w*10..*w*18; *w*20..*w*28; *w*30..*w*38标记。其中*w*00, *w*10, *w*20, *w*30, 的字长为6-bit，内容分别指示置换序号(3-bit)和置换起点( 3-bit )。

*wi,j* (*i* = 0..3; *j* = 1..8) 的字长为5比特，指示组合矩阵A的行坐标。列坐标则经置换变换。

置换表大小均为8×8。矩阵的32列分为4组，置换表负责4次8列的置换。假设给定置换表是：

置换表

(0)(1)(2)(3)(4)(5)(6)(7)

[0] 7 4 2 3 5 1 6 7

[1] 4 6 3 5 0 7 2 3

[2] 6 0 7 6 4 3 7 5

[3] 1 2 6 1 7 0 5 6

[4] 2 7 0 2 3 5 1 0

[5] 0 1 3 7 6 2 4 4

[6] 5 3 1 0 2 4 3 2

[7] 3 5 5 4 1 6 0 1

置换表的列为置换序号，行为置换起点。在上表中，设：序号为3，起点为1为例：

输入序列 h g f e d c b a

Disk置换  3 4 0 7 2 1 6 5

0 7 6 5 4 3 2 1

输出序列 f e b a g h d c

经置换后的列坐标，重新用*ti,j*表示列坐标(*i* = 0..3, *j* = 1..8)。

Alice的组合私钥，由KMC计算：



Alice的组合公钥，由依赖方计算：

# 

密钥分别用下式表达私钥和公钥的生成：

HASHHkey(Bob)=i,j ∑(ri,j)=*bob*

HASHHkey(Bob)=i,j ∑(Ri,j)=*BOB*

其中，公钥用大、斜体标记，私钥用小、斜体标记。

# **4 数字签名**

CPK所用数字签名协议，是根据防量子穷举攻击的要求，将现行ECDSA改造而成的。

Alice的签名函数：

SIG*alice* (h) =( *s*, *c* )

其中，SIG是签名函数，*alice*是组合私钥，h是签名对象，如实体标识、日时组或数据的Hash码等，*s*是签名码，*c*是核对码。

Alice签名：选择一个随机数*k*（0<*k*<*n*），

*k* G= (*x*1, *y*1)

*c* = (*x*1+*y*1)2mod 2*m*

*s* = *k*-1 (h + *c alice* ) mod *n*

其中，2*m*用于校验码长度的选择。

Alice发送：sign = ( *s*, *c* )

Bob的验证：VER*ALICE*(*s* ) = *c*’

验证过程：Bob根据Alice的标识计算组合公钥：Alice→*ALICE*；

Bob根据签名码sign = (*s*, *c*) 计算

*s*-1hG+ *s-1 c ALICE*= (*x*1’, *y*1’)

*c*’= (*x*1’+*y*1’)2 mod 2*m*

如果*c* = *c*’, 签名被认可。

其中，VER是验证函数，*ALICE*是组合公钥。

# **5 密钥加密**

CPK所用密钥加密协议，满足防止量子计算攻击的要求，具体如下：

Alice给Bob的加密函数：

ENC*BOB*(key) =β

Ekey(data) = code

其中，ENC是非对称加密函数，*BOB*是对方公钥，*r*是随机数。

Alice加密：计算Bob的组合公钥：Bob→*BOB*

Alice选择随机数*r*，计算

*r BOB* = β

*r*G= (*x*1, *y*1)

key = (*x*1+y1)2 mod 264

Ekey (data) = code

Alice将{code, β}发送给Bob

Bob脱密：

DEC*bob*(β) = key

Dkey(code) = data

其中，DEC是非对称脱密函数，*bob*是自己的私钥。

Bob的脱密过程：用自己的组合私钥*bob*计算出脱密密钥

(*bob*)-1 β = *r*G = (*x*1, *y*1)

key *=* (*x*1+y1)2 mod 264

# **6 安全性证明**

CPK的安全涉及三个方面：体制的安全性，系统密钥的安全性，个人密钥的安全性。

### 6.1 体制的安全性

从理论上讲，世上没有永远不可破的密码，而且也没有必要。体制的安全性，主要防止解剖分析、复制分析、动态分析、静态分析等。为了增大解剖分析的难度，一般采用硬件实现体制，这只是增加分析的成本而已。体制的安全性并不是量子时代的新问题，而是从机械作业、电子作业、计算机作业时代的老问题。量子计算机与电子计算机比较，最大的差别就是运算速度，将过去不可能的穷举变得可行，对密码的性质不引起实质性的变化。

密码或公钥，均由体制和密钥组成，体制是不变的固定因素，只有密钥是可变的活跃因素。体制的弱点在于固定性上，不易抗衡各种分析。体质的更换是很麻烦烦的事，因此一般用下去，不更换。网络化自动管理，为体质的更换提供了方便。这是互联网、物联网时代提供的新的机遇。

穷举是数学问题，只能用数学的方法才能解决。任何穷举只有存在一个判别条件时才有意义，否则没有意义。

一是公钥和私钥都要保密，而只有基于标识的公钥体制才有可能做到公钥和私钥同时保密。当一定密钥长度能够阻止电子计算机的穷举攻击时，采用公开公钥的方法是解决密钥分发难题的最简单方法。但在量子计算面前，只要公钥一公开，就为私钥的穷举提供判别条件，抗不住量子计算的穷举。例如，在ECC公钥体制*aG*=*A*中，*a*是私钥，*A*是公钥，*G*是椭圆曲线的基点，基点*G*是已知因素，只要再把公钥*A*公开，那么就可以穷举私钥*a*，并能得到唯一解。反过来，如果将公钥*A*能保密，*a*就失去了穷举的判别依据，穷举就失去意义。在基于标识的公钥体制中，标识代表实体，且具有公认性。标识的公开，为不公开公钥提供了可能性。

二是在签名协议或密钥传递协议中，公钥和私钥必须以不可穷举的和数形态出现，以简单方程 (*a* *b*) mod 13 = 7为例， *a*和*b*是未知数。一眼可看出方程就有无穷解，与穷举能力无关。

### 6.2 系统密钥的安全性

CPK私钥是整数的线性组合，在私钥之间存在线性关系，可以列出方程组。但是这种线性关系没有任何意义，因为组合矩阵不提供满秩方程，求不出方程的唯一解。

### 6.3 个人密钥的安全

对个人密钥安全的最大威胁将是穷举攻击。只要公钥一公开，就成为穷举私钥的判别依据。因此，公钥和私钥都要作为秘密变量，如在以私钥进行的签名与脱密作业中私钥不能暴露在外面一样，当以公钥进行的验证与加密作业中公钥也不能以独立变量暴露在外面。在数字签名协议和密钥传递协议中的私钥或公钥，应以合成数的形式出现，形成无穷解方程。

1）在数字签名函数中的个人密钥

数字签名的验证过程如下：

VER*ALICE*(*s* ) = *c*’

Bob根据Alice的标识计算组合公钥：Alice→*ALICE*； （1）

Bob根据签名码sign = (*s*, *c*) 计算

*s*-1hG+ *s-1 c ALICE*= (*x*1’, *y*1’) （2）

*c*’= (*x*1’+*y*1’)2 mod 2*m* （3）

如果*c* = *c*’, 签名被认可。在上（1）式中暴露了公钥，为私钥的穷举提供了判别条件，实际上这样求出来的公钥还不是真正的公钥，真真的公钥还差一步运算。经运算算出来的公钥直接进入下一步合成运算，很难孤立出来。式（2）中的公钥收到两个未知数的保护。在(3)式中，只有*x*1是未知因素，可以穷举。假设密钥长度为*n =* 192，验证码长度为*m* = 40，当穷举*x*1时得到2192/240 = 2152个符合*c*的结果，并通过(2)式求出2152个可能的*s*和*alice*。私钥*alice*与随机数*s*形成自变量和应变量的关系。

另外，在本例中，当*m* = 40时，其验证错判的率概仅为1/240，且能大大缩短签名长度(从48个字节缩短到32个字节)。

2）密钥传函数中的个人密钥

在数据加密中暴露的因素是{ β, code }. 其中

*r BOB* = β (4)

*r G* = (*x*1, *y*1) (5)

key = (*x*1+*y*1)2 mod 264  (6)

Ekey (data) = code (7)

在(4)式中，*β*是给定因素，是一个随机数*r*和公钥*BOB*的乘积，

β= *r BOB* = *r bob* G = *v* G,

由*v* G可求出*v*，而*v*是*r*和*bob*乘积，形成无穷解方程。

在(7)式中，假设明文可成为判别的依据，加密密钥key可被穷举的情况下，仍可提供2192/264 =2128个可能的*x*1，在(5)式中，可求出2128个可能的*r*, 回到(4)式，虽然将随机数*r*的范围缩小到2128，但无助于确定私钥*bob*或公钥*BOB*。

**小结**

CPK将密钥生成和分发有机结合起来，简化了密钥管理的复杂性。数字签名实现了短签名，可以直接应用于标识真实性证明和实体真实性的证明，密钥加密可以直接应用于数据加密系统。CPK可灵活应用于多样化需求，可以用芯片实现，也可以用软件实现，也可以将软硬件结合起来，芯片中只保留必要的参数，以低廉的成本为国家提供网络安全，为公众提供交易安全。

###### 参考文献

[1] The Whitehouse, Washington, National Strategy for Trusted Identities in Cyberspace, Enhancing Online Choice, Efficiency, Security, and Privacy, Apr. 2011

[2] Adi Shamir, Identity-Based Cryptosystems and Signature Schemes [C], Advances in Cryptology, CRYPT 84, volume 196 of LNCS, Springer-Verlag, 198447-53

[3] Xianghao Nan, Identity Authentication, Technical Basis of Cyber Security, Publishing House of Electronics and Industry, Jun. 2011

[4] M. Brurros, M. Abadi, R, Needham, A logic of Authentication, ACM Trans, on Computer systems, 1990

[5] 南湘浩、陈钟，网络安全技术概论，国防工业出版社，2003.7

[6] Mingyuan Yu, Xianping Huang, Li Jiang, Ronghua Liang, Combined Public Key Cryptosystem Based on Conic Curves over the Ring **Z**n, 2008 International Conference on Computer Science and Software Engineering

[7] Standard for Efficient Cryptography. SEC1: Elliptic Curve Cryptography 2000

[8] Standard for Efficient Cryptography. SEC2: Elliptic Curve Cryptography 2000

[9] National Institute of Standards and Technology, INST PUB 186, Digital Signature Standards, U.S. Department of Commerce 1994

[10] W. Diffie, M. E. Helman, New Direction of Cryptography [J]. IEEE Trans IT 1976, 22(6): 644-655

[11] ElGamal T, A Public key Cryptosystem and a Signature Scheme[J], IEEE Transactions on Information Theory, 1985, 31(4): 469-472

[12] Xianghao Nan, CPK Crypto-system and Identity Authentication, Publishing House of Electronics and Industry, Sep. 2012.

**注：CPK各版本的简要说明：**

**CPK v1.0**: 由一个（32x32）的组合矩阵构成，体制简练，但存在公开问题：即在私钥间存在线性关系，需要继续改进。

专家会：CPK解决了基于标识的密钥管理难题，解决了当前进程认证的难题，具有重大创新意义和广阔的应用前景。

新华社：就CPK报送了《我发明“电子身份证”，引起国际关注》的内参稿，对此，胡锦涛做了批示：“好东西就要用”。

国际密码94’主席James Hughes：”CPK将基于标识的公钥体制向前推进了一大步，并提供公钥密码梦寐以求的所有好处。”

国家密码管理局：“CPK理论上不具备可证明安全性，还存在共谋危险”，并在全国掀起了封杀活动。

**CPK v2.0**: 将组合矩阵大小改为（*h*x32），*h*>32，即h可以增加到几百万，仍未获批准。为了消除私钥间的线性关系，增设了随机密钥对，称TF-CPK，随获国家密码管理局的批准，但由于没有对随机密钥的第三方公证，存在替换攻击。

**CPK v3.0**: 为了防止替换攻击，增加了用标识密钥对随机密钥的公证，其芯片通过了国密局的测试。但v3.0本版本不能进行密钥传递，进而仍不能应用于数据加密，失去了CPK原有特点，不得不被自我否定。

**CPK v4.0**: 国密局的结论受到一个外国教授的嘲讽，感到丢脸，于是彻底否定了国密局的结论，并取消了增设的密钥对序列（TF），恢复了CPK本来的面目，即恢复了数字签名和密钥加密功能；

**CPK v5.0**: 将随机密钥对有限序列改为组合矩阵B，变成双矩阵；

**CPK v6.0**: 将密钥生成，数字签名，密钥传递等过程均以函数化(模块化)实现；取消了组合矩阵B

**CPK v7.0**: 增设了列坐标的置换变换，

**CPK v8.0**: 对签名协议和密钥加密协议采取了防量子计算穷举的措施。并将签名名长度从48字节缩短到29字节。实现了系统和私钥均用网上的自动分发。

**CPK v9.0**: 解决了在开放环境中私钥存储技术，并应用于CPK-chip或CPK-soft。