(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2009 年3 月5 日 (05.03.2009)

PCT

(10) 国际公布号
WO 2009/026771 A1

(51) 国际专利分类号:
H04L 9/32 (2006.01) *H04K 1/00* (2006.01)

(21) 国际申请号: PCT/CN2007/070628

(22) 国际申请日: 2007 年9 月5 日 (05.09.2007)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
200710120763.X
2007 年8 月24 日 (24.08.2007) CN

(71) 申请人 (对除美国外的所有指定国): **管海鹰 (GUAN, Haiying)** [CN/CN]; 中国北京市海淀区复兴路20号东区48楼1101号, Beijing 100036 (CN)。

(71) 申请人及

(72) 发明人: **管海明 (GUAN, Haiming)** [CN/CN]; 中国北京市海淀区万寿路6号3号楼1单元201, Beijing 100036 (CN)。

(74) 代理人: **北京集佳知识产权代理有限公司 (UNITALEN ATTORNEYS AT LAW)**; 中国北京市朝阳区建国门外大街22号赛特广场7层, Beijing 100004 (CN)。

(81) **指定国** (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) **指定国** (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA,

[见续页]

(54) Title: THE METHOD FOR NEGOTIATING THE KEY, ENCRYPTING AND DECRYPTING THE INFORMATION, SIGNING AND AUTHENTICATING THE INFORMATION

(54) 发明名称: 密钥协商的方法、加/解密的方法及签名/验证的方法

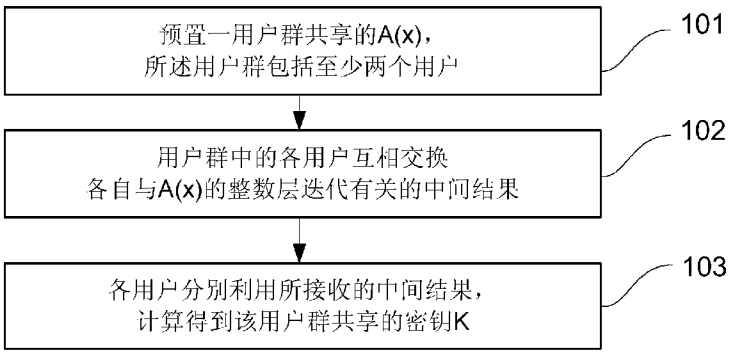

图 1 / Fig. 1

101 THE SHARED A(X) OF ONE USER GROUP IS PRE-SET, AND THE USE GROUP INCLUDES AT LEAST TWO USERS
102 THE USERS OF THE USER GROUP INTERCHANGE THE MIDDLE RESULTS OF THE ITERATION OF THE A(X)
103 THE USERS CALCULATE THE SHARED KEY K ACCORDING TO THE MIDDLE RESULTS

(57) Abstract: The method for negotiating the key, encrypting and decrypting the information, signing and authenticating the information includes the following steps, Step 1, the shared A(x) of one user group is pre-set, and the use group includes at least two users, the A(x) is the nonlinear function group, in which the vector X of n variables is transformed to the vector Y of n variables, $y=(y_1,..., y_n)=A(x)=(A_1(x_1,..., x_n),...,A_n(x_1,..., x_n))$, where n>1. As to A(x), the number of the coefficients of x which is not equal to 0 in the s-layered iteration $A^{(s)}(x)$ is unchanged, where the s is the integer. When B(x)=A(A(x)), then A(b(x))=B(A(x)). Step 2, the users of the user group interchange the middle results of the iteration of the A(x). Step 3, the users calculate the shared key K according to the middle results. So the complexity of cryptography and performance of anti-attack are improved.

[见续页]

WO 2009/026771 A1

SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧洲 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。

本国际公布：
— 包括国际检索报告。

(57) 摘要：

本发明提供了一种密钥协商方法、加/解密方法及签名/验证的方法，其中的密钥协商方法可以包括：步骤 1，预置用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组 $\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(A_1(x_1, \ldots, x_n), \ldots, A_n(x_1, \ldots, x_n))$其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；步骤 2，所述用户群中的各用户互相交换各自与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果；步骤 3，各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。本发明具有独特的编码风格和很强的抗攻击能力，显著增强密码算法的规模和复杂性，提高算法空间和安全性。

密钥协商的方法、加/解密的方法及签名/验证的方法

本申请要求于 2007 年 8 月 24 日提交中国专利局、申请号为200710120763.X、发明名称为“密钥协商的方法、加/解密的方法及签名/验证的方法”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明涉及信息安全与密码技术领域，特别是涉及一种能够完成密钥协商、对数据消息加密/解密以及签名/验证的公钥密码体制。

背景技术

密码技术是研究加密和解密变换的一门科学技术。通常情况下，人们将可懂的文本称为明文；将明文变换成的不可懂的文本称为密文。把明文变换成密文的过程叫加密；其逆过程，即把密文变换成明文的过程叫解密。这种加密或解密变换是由密钥来控制的。在开放环境下使用的密码系统应满足以下基本要求：

保密性：保证信息不被泄漏给非授权的用户；

完整性：保证信息不被任意或蓄意地修改；

抗抵赖性：防止个人或实体通过销毁证据来否认曾经发布过的信息，以证明某类事件确实曾经发生过。

公钥密码是解决上述的保密性、完整性、抗抵赖性的关键技术。其正式诞生的标志是1976年W.Diffie和M.Hellman发表的《密码学的新方向》（W. Diffe, M. E. Hellman, “New direction in cryptography”, IEEE Trans., 1976, 22, 644-654）。公钥密码使用一个公钥和一个私钥，公钥可以公开传递，但相关的私钥是保密的。只有使用私钥才能解密用公钥加密的数据、并对数据进行签名，公钥的作用则是对信息进行加密、以及验证签名的正确性。公钥密码还可以实现密钥协商协议，即两个用户在事先没有任何秘密约定的条件下，在完全公开的信道上，建立双方共享的密钥。

目前被公认为有较强的安全性、已广泛应用的公钥密码编码方案，按照所基于的数学难题分类，只有以下三种：

一是 RSA 体制。由 Rivest、Shamir 和 Adleman 在 1978 年共同发明的公

钥密码体制（R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures public-key cryptosystems”, Communications of the ACM, 21(1978), 120-126），其安全性基于大整数因子分解问题。

二是 DH 体制。由 Diffie 和 Hellman 在 1976 年发明的密钥协商协议，以及由 ElGamal 在 1985 年提出的 ElGamal 加密和数字签名方案（T. ElGamal, “A public key cryptosystem and signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, 31(1985), pp.469-472.），其安全性基于有限域的乘法群上的离散对数问题。这种类型的算法还包括美国数字签名标准 DSS（美国联邦信息处理标准 FIPS 186）等。

三是 ECC 体制。由 Miller 和 Koblits 在 1985 年各自独立地发明的椭圆曲线公钥密码体制（V. S. Miller, “Use of elliptic curve in cryptography”, CRYPTO’85, Springer-Verlag, 1986, pp.417-426.）（N. Koblitz, “Elliptic curve cryptosystems”, Mathematics of Computation, v.48, n.177, 1987, pp.203-209.），其安全性基于椭圆曲线群的离散对数问题。这种类型的算法还包括超椭圆曲线公钥密码体制（N. Koblitz, “Hyperelliptic cryptography”, J.of Crypto., 1989, 1(3), pp.139-150.）。

值得注意的是，RSA、DH、ECC 被大量使用，但其安全性都没有得到理论证明，主要是由于现实需要（签名、识别、支付、密钥管理等），在苦于没有其它替代技术的情况下，不得不用。但是，由于上述三种公钥密码体制的安全性并没有得到数学理论的证明，所以不排除这样一种可能：经过几十年的分析研究，实际已经有人找到了破译它们的有效方法，只不过这个事实没有公开而已。

并且，随着量子计算机的研究进展，支持上述三种公钥密码体制被破译的可能性大大增加。例如，由 Shor 在 1994 年发明的 Shor 算法（P. W. Shor, “Algorithms for quantum computation: Discrete log and factoring”, Proceedings of the 35th Symposium on Foundations of Computer Science, 1994, pp.124-134.），能以多项式时间攻破所有的能够转换成广义离散傅立叶变换的公钥密码。

为此，构建具有更大的算法空间和更强安全性的公钥密码体制，具有重要意义。本发明便是基于这种思想而完成的研究结果。

发明内容

本发明所要解决的技术问题是运用保形迭代变换的方法，提供一种公钥密码体制编码方法和装置，以实现具有更大的算法空间和更强安全性的密钥协商、加解密和数字签名的技术方案。

为了解决上述问题，依据本发明的实施例，公开一种密钥协商的方法，包括：

步骤 1，预置用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2，所述用户群中的各用户互相交换各自与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果；

步骤 3，各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

优选的，当该用户群仅包括两个用户时，所述步骤 2 进一步包括：

第一用户选择整数 k_1，计算第一中间结果，并传递至第二用户；所述第一中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_1 层迭代有关；

第二用户选择整数 k_2，计算第二中间结果，并传递至第一用户；所述第二中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_2 层迭代有关。

进一步，可以依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；采用该伪随机序列的种子，标识该 $\mathbf{A}(\boldsymbol{x})$。

依据本发明的另一实施例，公开了一种用于编码和译码数字消息的方法，包括：

步骤 1，预置加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若

$\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x}))=\mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2、选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

步骤 3、加密端选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，然后利用该中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

步骤 4、解密端利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，然后利用该中间密钥对加密结果进行解密。

优选的，通过以下步骤建立私钥：

预置λ个私钥表$L_1, \ldots, L_\lambda$以及对应的公钥表$\mathbf{G}_1, \ldots, \mathbf{G}_\lambda$，分布在$\lambda$个密钥分配中心；

依据预置规则，根据用户的身份ID获得指向多个私钥表的指针；

分别从所指向的多个私钥表中各获取一个或者多个私钥分量，组合得到该用户的私钥。

依据本发明的另一实施例，公开了一种用于数字签名及验证的方法，包括：

步骤 1，预置签名端和验证端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x}))=\mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2、选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

步骤 3、签名端选择整数 t，依据预置规则将待签名数据变换为与 t、私钥 k 相关的中间消息，然后传送包含中间消息和 t 的变换结果的数字签名至验证端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

步骤 4、验证端利用 t 的变换结果、待签名数据、中间消息、公钥和 $\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则，如果满足，则该数字签名验证通过。

依据本发明的另一实施例，公开了一种密钥协商的系统，包括：

共享单元，用于存储用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相

比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

中间结果交换单元，连接所述用户群中的各用户端，用于传递所述用户群中的各用户与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果至其他用户；

密钥计算单元，位于所述用户群中的各用户端，用于针对各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

依据本发明的另一实施例，还公开了一种用于编码和译码数字消息的系统，包括：

共享单元，用于存储加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

加密单元，位于加密端，用于选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，利用该中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

解密单元，位于解密端，用于利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，利用该中间密钥对加密结果进行解密。

依据本发明的另一实施例，还公开了一种用于数字签名及验证的系统，包括：

共享单元，用于存储签名端和验证端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

签名单元，位于签名端，用于选择整数 t，依据预置规则将待签名数据变换为与 t、私钥 k 相关的中间消息，传送包含中间消息和 t 的变换结果的数字签名至验证端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

验证单元，位于验证端，用于利用 t 的变换结果、待签名数据、中间消息、公钥和 $\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则，如果满足，则该数字签名验证通过。

与现有技术相比，本发明具有以下优点：

本发明创造性的提出：运用基于域上的或环上的多变元非线性保形迭代变换的层数问题，构建公钥密码体制；该公钥密码体制可实现密钥协商、加密和数字签名，具有独特的编码风格和很强的抗攻击能力，使密码算法的规模和复杂性获得显著增强，以解决现有技术存在的算法空间小、安全性不够等问题。

附图说明

图 1 是本发明一种密钥协商的方法实施例的步骤流程图；

图 2 是本发明一种建立非线性函数组 A(x)的方法实施例的步骤流程图；

图 3 是本发明另一种建立非线性函数组 A(x)的方法实施例的步骤流程图；

图 4 是本发明一种用于编码和译码数字消息的方法实施例的步骤流程图；

图 5 是本发明一种用于数字签名及验证的方法实施例的步骤流程图；

图 6 是本发明一种数字签名数据流的示意图；

图 7 是本发明一种签名验证数据流的示意图；

图 8 是本发明的安全性所基于的数学难题－基于多变元非线性保形迭代变换的迭代层数问题的示意图 1；

图 9 是本发明的安全性所基于的数学难题－基于多变元非线性保形迭代变换的迭代层数问题的示意图 2。

具体实施方式

为使本发明的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本发明作进一步详细的说明。

本发明属于信息安全产品的范畴，主要应用于网络信任系统，例如证件、银行、手机、互联网、电子商务、电子政务、物流、网络监控、权力控制、资金转移、交易、数据加密等环节。

应用本发明所需的硬件环境属于本领域技术人员所熟知的知识。例如：计

算机、网络设备、手持设备或便携式设备、可编程的消费电子设备、智能卡、单片机、专用数字信号处理芯片、多处理器系统、包括以上任何系统或设备的分布式计算环境等等。

下面对本发明可能涉及的一些术语进行简单解释：

密码：通常可理解为进行信息加密和解密变换的算法。它的基本目的是伪装信息，使局外人不能理解信息的真正含义，而局内人能够理解伪装信息的本来含义。

密钥：在执行密码算法的过程中，唯一能控制明文与密文之间进行有效变换的关键参数。

公钥密码体制：公钥密码体制使用两个密钥——一个公开密钥（简称：公钥）和一个私人密钥（简称：私钥）。公钥和私钥在数学上是相关的，但由公钥计算出私钥是困难的。公钥可在通信双方之间公开传递，也可以像电话号码本一样公开发布，私钥则由授权用户自己秘密保管。任何人从某个用户的名字就能查到它的公钥，因而可以给这个用户发送加密消息。只有授权用户自己才能用他的私钥完成解密。

公钥密码体制还提供了**数字签名及认证**的能力：授权用户能用他的私钥对信息进行签名（相当于上述用私钥解密的过程）；其他用户由于不掌握私钥而不能进行签名，但能用该用户的公钥验证签名的正确性（相当于上述用公钥加密的过程）。

密钥协商协议（key agreement protocol）两个或者多个用户在事先没有任何秘密约定的条件下，在完全公开的信道上，建立双方或多方共享的密钥。

域（finite field）：是一种具体而又形象的数学结构，可以通俗地理解为能进行加减乘除四则运算的有限个元素的集合。（通常记做 **F**，当域的元素数量为素数 p 时，记做有限域 $\mathbf{F}_p$。）

有限域上的多项式（polynomial）：可以通俗地理解为当只有一个变元时：

$$f(x) = a_s x^s + a_{s-1} x^{s-1} + \ldots + a_0 x^0 \pmod{p}$$

其中x叫作变元，a_i叫作系数，$a_i x^i$叫作项，它们在0,…,p−1之间取值。当有多个变元时：

$$f(x_1, \ldots, x_n) = \sum_{\substack{i_1, \ldots, i_n \\ i_1 + \cdots + i_n \le s}} a_{i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n} \pmod{p}$$

如果多项式中的项的数量相对很少，叫做**稀疏多项式**；反之叫做**稠密多项式**。稠密多项式不仅有很高的次数，而且项的数量非常多，把它展开来表示需要占用很大的空间位置。

有限域上的有理分式（rational fraction）：可理解为两个多项式相除：

$$\frac{f(x_1,...,x_n)}{g(x_1,...,x_n)} \bmod p$$

除了 0 多项式以外的多项式的乘法逆为

$$(f(x_1,\ldots,x_n))^{-1 \pmod{p-1}} = (f(x_1,\ldots,x_n))^{p-2} \pmod{p}$$

但当 p 较大时，把上式展开需要巨大的存储空间，因此两个稀疏多项式相除（分母不为 0 多项式）的结果，通常是一个稠密多项式：

$$\frac{f(x_1,...,x_n)}{g(x_1,...,x_n)} = f(x_1,...,x_n)\cdot(g(x_1,...,x_n))^{p-2} \pmod{p}$$

有理函数（rational function）：可用两个多项式相除表示的函数。若其分母多项式的次数大于 0，则表示为有理分式，若其分母为 0 次多项式，则表示为多项式。

环（ring）：是一种数学结构，记做 **R**，可以通俗地理解为具有加法和乘法两种运算并满足乘法分配律的元素的集合。例如，由{0, 1, …m-1}组成的其元素数量为正整数 m 集合，以及在模 m 的意义上规定的加法和乘法，叫做整数剩余类环 $\mathbf{Z}_m$。

环上的多元多项式（polynomial），例如：

$$f(x_1,...,x_n) = \sum_{\substack{i_1,...,i_n \\ i_1+\cdots+i_n \le s}} a_{i_1\ldots i_n} x_1^{i_1}\ldots x_n^{i_n} \pmod{m}$$，m 不一定是素数

由于环的数学结构中没有定义除法运算，因此只能建立环上的多项式，而不能建立环上的有理分式。

参照图 1，示出了一种密钥协商的方法实施例，具体可以包括：

步骤 101，预置一用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1,\ldots,y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1,\ldots,x_n),\ldots,\mathrm{A}_n(x_1,\ldots,x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 102，所述用户群中的各用户互相交换各自与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果；

步骤 103，各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

本实施例可以通过所述用户群中的各用户互相交换隐藏在 $\mathbf{A}(\boldsymbol{x})$的迭代结果中的整数，而达到在公开信道上建立各方共享的密钥。密钥协商成功之后，就可以进行对称加密了。一般情况下，密钥协商的目的就是建立对称密码使用的密钥。通常的原因是：公钥加密速度太慢，一般先用公钥建立对称密码使用的密钥，再用对称密码以较快的速度完成加解密。这种方式的使用目的还包括无密钥保密通信，即每次通信都要临时进行密钥协商的保密通信，特点是不怕密钥被事先泄露，使得内部人员出卖密钥变的没有意义，因为公私钥方式毕竟还有一个私钥被事先泄露的问题。

所述的用户群可以包括两个或者两个以上的用户，当然，需要该用户群内的各个用户都互相交换信息，才能够建立整个群上共享的密钥。由于两个用户之间互相交换信息是多个用户互相交换的基础，并且多个用户之间的信息交换过程可以看作是用户两两之间交换的重复过程，所以下面都以两个用户为例进行说明。

优选的，当该用户群仅包括两个用户时，所述步骤 2 可以进一步细化包括：第一用户选择整数 k_1，计算第一中间结果，并传递至第二用户；所述第一中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_1 层迭代有关；第二用户选择整数 k_2，计算第二中间结果，并传递至第一用户；所述第二中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_2 层迭代有关。

在本发明的另一个优选实施例中，如果还包括以下步骤：建立该用户群共享的、变元数量大于 1 的向量 $\boldsymbol{q}$；并且，该用户群仅包括两个用户，则

所述步骤 102 进一步包括：第一用户选择整数 k_1，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_1 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_1 = \mathbf{A}^{(k_1)}(\boldsymbol{q})$，把计算结果 $\boldsymbol{d}_1$ 传递给第二用户；第二用户选择整数 k_2，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_2 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_2 = \mathbf{A}^{(k_2)}(\boldsymbol{q})$，把计算结果

$\boldsymbol{d}_2$传递给第一用户；

所述步骤103进一步包括：第一用户计算密钥$\boldsymbol{K}=(K_1,...,K_n)=\mathbf{A}^{(k_1)}(\boldsymbol{d}_2)$；第二用户计算密钥$\boldsymbol{K}=(K_1,...,K_n)=\mathbf{A}^{(k_2)}(\boldsymbol{d}_1)$；

其中，所述$\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k_1)}(\mathbf{A}^{(k_2)}(\boldsymbol{x}))=\mathbf{A}^{(k_1+k_2)}(\boldsymbol{x})$，这样才可以保证第一用户和第二用户计算得到的密钥$\boldsymbol{K}$相同。

在本发明的另一个优选实施例中，当该用户群仅包括两个用户时，

所述步骤102进一步包括：第一用户选择整数k_1，计算k_1层$\mathbf{A}(\boldsymbol{x})$的迭代：$\mathbf{B}_1(\boldsymbol{x})=\mathbf{A}^{(k_1)}(\boldsymbol{x})$，并把函数组$\mathbf{B}_1(\boldsymbol{x})$传递给用户2；第二用户选择整数$k_2$，计算$k_2$层$\mathbf{A}(\boldsymbol{x})$的迭代：$\mathbf{B}_2(\boldsymbol{x})=\mathbf{A}^{(k_2)}(\boldsymbol{x})$，并把函数组$\mathbf{B}_2(\boldsymbol{x})$传递给第一用户；

所述步骤103进一步包括：第一用户计算密钥$\boldsymbol{K}=\mathbf{B}_2{}^{(k_1)}(\boldsymbol{x})$；第二用户计算密钥$\boldsymbol{K}=\mathbf{B}_1{}^{(k_2)}(\boldsymbol{x})$；

其中，所述$\mathbf{A}(\boldsymbol{x})$进一步满足：若$\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k_1)}(\boldsymbol{x})$，则$\mathbf{B}^{(k_2)}(\boldsymbol{x})=\mathbf{A}^{(k_1k_2)}(\boldsymbol{x})$，从而可以保证第一用户和第二用户计算得到的密钥$\boldsymbol{K}$相同。

下面对如何建立合适的$\mathbf{A}(\boldsymbol{x})$进行简单介绍，当然，除了本发明公开的这些函数类型及其建立方法，实际中，还可能存在其他的$\mathbf{A}(\boldsymbol{x})$函数类型及建立$\mathbf{A}(\boldsymbol{x})$的方法，比如以指数幂方式出现的有限域或有限环上的函数；在此无法一一详述，仅描述本发明的优选实施方式。只要建立得到的$\mathbf{A}(\boldsymbol{x})$满足本发明的限定要求即可。步骤101中所述的“预置”可以包括：实时建立、预先建立或者他人建立等多种方式。

在下面的描述中，本发明给出了三种类型的$\mathbf{A}(\boldsymbol{x})$的建立方法。设$n>1$，$\mathbf{F}$为规定的域，$\mathbf{R}$为规定的环，$\boldsymbol{x}=(x_1,\ldots,x_n)$，$\boldsymbol{y}=(y_1,\ldots,y_n)$，$\boldsymbol{z}=(z_1,\ldots,z_n)$，$x_i, y_i, z_i\in\mathbf{F}$或$\mathbf{R}$；随机选择一个$n$元非线性保形迭代变换：$\boldsymbol{y}=\mathbf{A}(\boldsymbol{x})$，则可以从下面的三种建立方法中选择。

第一种类型

参照图2，第一种类型的n元非线性函数组$\mathbf{A}(\boldsymbol{x})$可以通过以下步骤建立：

步骤201、预置$\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由$n$个域$\mathbf{F}$上的$n$元有理分式函数组成，其每个有理分式函数中的分子、分母均为关于$(x_1,\ldots,x_n)$的线性多项式，其分母多项式相同；

步骤202、接收$\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量$n$

和变元的数据长度；

步骤 203、生成 **A**(***x***)中的每个项的系数；

步骤 204、按照预置结构，输出得到的 **A**(***x***) 。

具体而言，第一种类型的 **A**(***x***)由 *n* 个 **F** 上的 *n* 元有理分式函数组成：

$\boldsymbol{y} = (y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x}) = (\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$，其中：

$$y_i = \mathrm{A}_i(x_1, \ldots, x_n) = \frac{a_{i0} + a_{i1}x_1 + \ldots + a_{in}x_n}{a_{00} + a_{01}x_1 + \ldots + a_{0n}x_n},$$

$$a_{ij}, x_i, y_i \in \mathbf{F}，\ 1 \leqslant i \leqslant n，\ 0 \leqslant j \leqslant n，$$

其有理分式中的分子、分母均为线性多项式，分母相同，该 **A**(***x***)的特征是：把其代入自身并展开、化简，

$\boldsymbol{z} = (z_1, \ldots, z_n) = \mathbf{A}(\mathbf{A}(\boldsymbol{x})) = (\mathrm{A}_1(y_1, \ldots, y_n), \ldots, \mathrm{A}_n(y_1, \ldots, y_n))$，其中：

$$z_i = \frac{a_{i0} + a_{i1}\frac{a_{10}+a_{11}x_1+\ldots+a_{1n}x_n}{a_{00}+a_{01}x_1+\ldots+a_{0n}x_n} + \ldots + a_{in}\frac{a_{n0}+a_{n1}x_1+\ldots+a_{nn}x_n}{a_{00}+a_{01}x_1+\ldots+a_{0n}x_n}}{a_{00} + a_{01}\frac{a_{10}+a_{11}x_1+\ldots+a_{1n}x_n}{a_{00}+a_{01}x_1+\ldots+a_{0n}x_n} + \ldots + a_{0n}\frac{a_{n0}+a_{n1}x_1+\ldots+a_{nn}x_n}{a_{00}+a_{01}x_1+\ldots+a_{0n}x_n}} = \frac{b_{i0} + b_{i1}x_1 + \ldots + b_{in}x_n}{b_{00} + b_{01}x_1 + \ldots + b_{0n}x_n},$$

$$\text{满足：} \begin{cases} b_{ij} \neq 0, & \text{for } a_{ij} \neq 0 \\ b_{ij} = 0, & \text{for } a_{ij} = 0 \end{cases};$$

第二种类型

参照图3，第二种类型的*n*元非线性函数组**A**(***x***)可以通过以下步骤建立：

步骤 301、预置 **A**(***x***)的结构：**A**(***x***)由 *n* 个域 **F** 上的 *n* 元有理函数组成，其含有关于$(x_1, \ldots, x_n)$的大于 1 次的项；当 $\mathrm{A}_i(x_1, \ldots, x_n)$的分母为 0 次多项式时，该有理函数为多项式；当 $\mathrm{A}_i(x_1, \ldots, x_n)$的分母为大于 1 次的多项式时，该有理函数为有理分式；

步骤 302、接收 **A**(***x***)的相关技术指标参数，所述指标参数包括变元数量 *n*、变元的数据长度和最高的非线性次数；

步骤 303、依据所述指标参数和预置结构，生成一个 **A**(***x***)的表示形式，该 **A**(***x***)中的不为零的系数用变元符号表示；

步骤 304、将 **A**(***x***)代入自身并执行展开、化简的数据处理：**B**(***x***) = **A**(**A**(***x***))；

步骤 305、针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，生成关于

这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

步骤 306、判断该方程组是否有解，如果没有解，则返回步骤 303；如果有解，则计算得到该方程组的一组解，并作为 $\mathbf{A}(\boldsymbol{x})$中的系数的值，代入步骤 303 生成的 $\mathbf{A}(\boldsymbol{x})$的表示形式；

步骤 307、输出所得到的 $\mathbf{A}(\boldsymbol{x})$。

具体而言，第二种类型的 $\mathbf{A}(\boldsymbol{x})$由 n 个 $\mathbf{F}$ 上的 n 元有理函数组成：

$\boldsymbol{y} = (y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x}) = (\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$，其中：

$$y_j = \mathrm{A}_j(x_1, \ldots, x_n) = \frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{j1, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{j0, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}},$$

$$a_{j0, i_1 \ldots i_n}, a_{j1, i_1 \ldots i_n}, x_i, y_j \in \mathbf{F}, \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant n,$$

其含有关于 $x_1, \ldots, x_n$ 的 l_1 次的项，$l_1 > 1$，当分母多项式为常数时该有理函数为多项式，该 $\mathbf{A}(\boldsymbol{x})$的特征是：把其代入自身并展开、化简，

$\boldsymbol{z} = (z_1, \ldots, z_n) = \mathbf{A}(\mathbf{A}(\boldsymbol{x})) = (\mathrm{A}_1(y_1, \ldots, y_n), \ldots, \mathrm{A}_n(y_1, \ldots, y_n))$，其中：

$$z_j = \frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{j1, i_1 \ldots i_n} \left(\frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{11, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{10, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}} \right)^{i_1} \cdots \left(\frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{n1, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{n0, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}} \right)^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{j0, i_1 \ldots i_n} \left(\frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{11, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{10, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}} \right)^{i_1} \cdots \left(\frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{n1, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_1}} a_{n0, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}} \right)^{i_n}}$$

$$= \frac{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_2}} b_{j1, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}}{\sum\limits_{\substack{i_1, \ldots, i_n \\ i_1+\ldots+i_n \le l_2}} b_{j0, i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n}},$$

$$\text{满足：}\begin{cases} l_2 = l_1 > 1 \\ b_{jk,i_1\ldots i_n} \neq 0, \quad \text{for } a_{jk,i_1\ldots i_n} \neq 0 \\ b_{jk,i_1\ldots i_n} = 0, \quad \text{for } a_{jk,i_1\ldots i_n} = 0 \end{cases};$$

第三种类型

第三种类型的n元非线性函数组$\mathbf{A}(\boldsymbol{x})$可以通过以下步骤建立，由于流程步骤非常相似，因此也可以参见图3。

步骤 a、预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个环 $\mathbf{R}$ 上的 n 元多项式组成：其含有关于$(x_1, \ldots, x_n)$的大于 1 次的项；

步骤 b、接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n、变元的数据长度和最高的非线性次数；

步骤 c、依据所述指标参数和预置结构，生成一个 $\mathbf{A}(\boldsymbol{x})$的表示形式，该 $\mathbf{A}(\boldsymbol{x})$中的不为零的系数用变元符号表示；

步骤 d、将 $\mathbf{A}(\boldsymbol{x})$代入自身并执行展开、化简的数据处理：$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$；

步骤 e、针对 $\mathbf{B}(\boldsymbol{x})$与 $\mathbf{A}(\boldsymbol{x})$对比而新出现的每一个关于 $\boldsymbol{x}$ 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

步骤 f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到该方程组的一组解，并作为 $\mathbf{A}(\boldsymbol{x})$中的系数的值，代入步骤 c 生成的 $\mathbf{A}(\boldsymbol{x})$的表示形式；

步骤 g、输出所得到的 $\mathbf{A}(\boldsymbol{x})$。

具体而言，第三种类型的 $\mathbf{A}(\boldsymbol{x})$由 n 个 $\mathbf{R}$ 上的 n 元多项式函数组成：

$\boldsymbol{y} = (y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x}) = (\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$，其中：

$$y_j = \mathrm{A}_j(x_1, \ldots, x_n) = \sum_{\substack{i_1, \ldots, i_n \\ i_1 + \ldots + i_n \leq l_1}} a_{j,i_1\ldots i_n} x_1^{i_1} \ldots x_n^{i_n},$$

$$a_{j,i_1\ldots i_n}, x_i, y_j \in \mathbf{R}，\ 1 \leqslant i \leqslant n，\ 1 \leqslant j \leqslant n$$

其含有关于 $x_1, \ldots, x_n$ 的 l_1 次的项，$l_1>1$，该 $\mathbf{A}(\boldsymbol{x})$的特征是：把其代入自身并展开、化简，

$$z_j = \sum_{\substack{i_1,\ldots,i_n \\ i_1+\ldots+i_n \le l_1}} a_{j,i_1\ldots i_n} \left(\sum_{\substack{i_1,\ldots,i_n \\ i_1+\ldots+i_n \le l_1}} a_{1,i_1\ldots i_n} x_1^{i_1} \ldots x_n^{i_n} \right)^{i_1} \ldots \left(\sum_{\substack{i_1,\ldots,i_n \\ i_1+\ldots+i_n \le l_1}} a_{n,i_1\ldots i_n} x_1^{i_1} \ldots x_n^{i_n} \right)^{i_n}$$

$$= \sum_{\substack{i_1,\ldots,i_n \\ i_1+\ldots+i_n \le l_2}} b_{j,i_1\ldots i_n} x_1^{i_1} \ldots x_1^{i_n} ,$$

$$\text{满足:} \begin{cases} l_2 = l_1 > 1 \\ b_{j,i_1\ldots i_n} \neq 0, \quad \text{for } a_{j,i_1\ldots i_n} \neq 0 \\ b_{j,i_1\ldots i_n} = 0, \quad \text{for } a_{j,i_1\ldots i_n} = 0 \end{cases} ;$$

实际上，为了达到很好的安全性，第二种和第三种类型的 **A**(***x***)应该满足：由 *s*、**A**(***x***)求 **B**(***x***)=**A**$^{(s)}$(***x***)容易，而由 **A**(***x***)、**B**(***x***)求 *s* 困难；或者，由 *s*、***q***、**A**(***x***)求 ***d*** =**A**$^{(s)}$(***q***)容易，而由 ***d***、***q***、**A**(***x***)求 *s* 困难，其中 ***q*** = (q_1, …, q_n)，***d*** = (d_1, …, d_n)，q_i, d_i∈**F** 或 **R**。

优选的，在第二种类型和第三种类型的 **A**(***x***)的建立过程中，可以存在很多的优化步骤，例如，在步骤 304 和 305 之间，或者在步骤 e 和 f 之间，还可以包括：将 **B**(***x***)与 **A**(***x***)对比，若 **B**(***x***)中的关于 ***x*** 的每一种新出现的项，都至少有两个，则执行步骤 305 或者步骤 e，否则返回到步骤 303 或者步骤 c。

原因在于，本发明的目的在于寻找满足保形迭代的具体函数，而如果存在一种新出现的项只有一个，则这个项的系数乘以任何不等于零的数都不可能等于零（若是环上的函数则等于零的可能性非常小），而若有两个以上的项，就可能让这些系数相加后等于零，从而在迭代后消去新增加的项。即可以达到初步过滤的效果，减少对解方程的调用次数，节省计算资源。

上述的用关于系数的不定方程组来建立 **A**(***x***)的方法，可以确保经过两层迭代后的函数规模不扩张，并能在很大的概率上保证所得到的 **A**(***x***)满足本发明的要求。当然，在本发明的优选实施例中，在步骤 306 和 307 之间，或者在步骤 f 和 g 之间，还可以包括更多的筛选步骤，用于对 **A**(***x***)作进一步的过滤，例如，验证经过 *k* 层迭代后的函数规模也不扩张，或者是否满足关于迭代运算的结合律等等。

需要说明的是，虽然上面的描述中将这三种类型的函数建立方法作为独立的三种方法进行介绍，但是本领域技术人员应该知悉，完全可以将其做在同一个执行流程中，只需要增加一选择步骤即可，在此不再详述。

下面参照图 3，通过具体的例子，对前述的建立过程进行详细描述，由于针对第二种类型和第三种类型的 **A(*x*)**的建立过程比较相似，所以合在一起作为一个具体例子进行介绍：

第一步，按照要求随意设置一个所期望的 A(*x*)的表示形式，把该 A(*x*)中的系数用变元符号表示：

通常可把元素数量为素数 p 的有限域 $\mathbf{F}_p$ 作为有限域 **F**，或把整数剩余类环 $\mathbf{Z}_N$ 作为有限环 **R**，但也可采用更加复杂的 **F** 或 **R**。

其 **A(*x*)**的表示形式由 n 个有理分式或者多项式组成，其函数中的系数用抽象的变量符号（诸如 $a_0, a_1, \ldots$等）表示。例如：

$$y_1 = \mathrm{A}_1(x_1, x_2) = (a_0 + a_1x_1 + a_2{x_1}^2 + a_3x_1x_2) \bmod p$$

$$y_2 = \mathrm{A}_2(x_1, x_2) = (b_0 + b_1x_1 + b_2{x_1}^2 + b_3x_1x_2) \bmod p$$

如何设置所期望的最佳的 **A(*x*)**的函数表示形式，已超出了本发明的内容范围，但又对本发明的实现效果产生明显的影响。在某种意义上说，这项工作往往需要凭直觉和经验来进行设计与分析，而不是完全依赖于严格的理论推导与证明。尤其对于复杂的非线性函数，有很多种选择，最好的办法是尝试不同的变换，直到获得所期望的函数形式。应把函数的每一层的具体算法、各层之间的关系，以及怎样把若干个简单函数组合成一个相对复杂的函数的推导过程，输入到 Mathematica 等软件，作为解方程的已知条件，以提高计算效率。

例如对于上述实施例，可先设置一个简单的可逆非线性变换：

$$h_1 = x_1，\ h_2 = (x_1x_2 + m{x_1}^2) \bmod p$$

再设置一个线性变换：

$$y_1 = (a_0 + a_1h_1 + a_2h_2) \bmod p，\quad y_2 = (b_0 + b_1h_1 + b_2h_2) \bmod p$$

然后把非线性变换代入到线性变换中，则所期望的 **A(*x*)**为：

$$y_1 = \mathrm{A}_1(x_1, x_2) = (a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2) \bmod p$$

$$y_2 = \mathrm{A}_2(x_1, x_2) = (b_0 + b_1x_1 + mb_2{x_1}^2 + b_2x_1x_2) \bmod p$$

其中 a_i、b_i、m 应理解为系数的因子。当然，也可以把 **A(*x*)**直接设置为：

$$y_1 = \mathrm{A}_1(x_1, x_2) = (a_0 + a_1x_1 + a_2{x_1}^2 + a_3x_1x_2) \bmod p$$

$$y_2 = \mathrm{A}_2(x_1, x_2) = (b_0 + b_1x_1 + b_2{x_1}^2 + b_3x_1x_2) \bmod p$$

然而这将导致数学软件无法获取该函数的结构信息，在进入到以下的第三步“判断方程组 **T** 是否有解”时可能会遇到计算困难。

第二步，把 **A(*x*)**代入自身，**推导出 *z* = A(*y*) = A(A(*x*)) = B(*x*)并展开：**

将 **B**(***x***)与 **A**(***x***)对比，若 **B**(***x***)中的关于 ***x*** 的每一种新出现的项的类型，都至少有两个这种类型的项，则 **A**(***x***)的表示形式满足要求；否则说明它不符合要求，应返回到第一步重新设置，即：

$$\boldsymbol{z} = (z_1, z_2)$$

$$z_1 = \mathrm{A}_1(y_1, y_2) = (a_0 + a_1y_1 + ma_2{y_1}^2 + a_2y_1y_2) \bmod p$$

$$= (a_0 + a_1(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2) + ma_2(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2)^2 + a_2(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2)\,(b_0 + b_1x_1 + mb_2{x_1}^2 + b_2x_1x_2)) \bmod p$$

$$= (a_0 + a_0a_1 + m{a_0}^2a_2 + a_0a_2b_0 + {a_1}^2x_1 + 2ma_0a_1a_2x_1 + a_1a_2b_0x_1 + a_0a_2b_1x_1 + ma_1a_2{x_1}^2 + m{a_1}^2a_2{x_1}^2 + 2m^2a_0{a_2}^2{x_1}^2 + m{a_2}^2b_0{x_1}^2 + a_1a_2b_1{x_1}^2 + ma_0a_2b_2{x_1}^2 + 2m^2a_1{a_2}^2{x_1}^3 + m{a_2}^2b_1{x_1}^3 + ma_1a_2b_2{x_1}^3 + m^3{a_2}^3{x_1}^4 + m^2{a_2}^2b_2{x_1}^4 + a_1a_2x_1x_2 + 2ma_0{a_2}^2x_1x_2 + {a_2}^2b_0x_1x_2 + a_0a_2b_2x_1x_2 + 2ma_1{a_2}^2{x_1}^2x_2 + {a_2}^2b_1{x_1}^2x_2 + a_1a_2b_2{x_1}^2x_2 + 2m^2{a_2}^3{x_1}^3x_2 + 2m{a_2}^2b_2{x_1}^3x_2 + m{a_2}^3{x_1}^2{x_2}^2 + {a_2}^2b_2{x_1}^2{x_2}^2) \bmod p$$

$$z_2 = \mathrm{A}_2(y_1, y_2) = (b_0 + b_1y_1 + mb_2{y_1}^2 + b_2y_1y_2) \bmod p$$

$$= (b_0 + b_1(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2) + mb_2(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2)^2 + b_2(a_0 + a_1x_1 + ma_2{x_1}^2 + a_2x_1x_2)\,(b_0 + b_1x_1 + mb_2{x_1}^2 + b_2x_1x_2)) \bmod p$$

$$= (b_0 + a_0b_1 + m{a_0}^2b_2 + a_0b_0b_2 + a_1b_1x_1 + 2ma_0a_1b_2x_1 + a_1b_0b_2x_1 + a_0b_1b_2x_1 + ma_2b_1{x_1}^2 + m{a_1}^2b_2{x_1}^2 + 2m^2a_0a_2b_2{x_1}^2 + ma_2b_0b_2{x_1}^2 + a_1b_1b_2{x_1}^2 + ma_0{b_2}^2{x_1}^2 + 2m^2a_1a_2b_2{x_1}^3 + ma_2b_1b_2{x_1}^3 + ma_1{b_2}^2{x_1}^3 + m^3{a_2}^2b_2{x_1}^4 + m^2a_2{b_2}^2{x_1}^4 + a_2b_1x_1x_2 + 2ma_0a_2b_2x_1x_2 + a_2b_0b_2x_1x_2 + a_0{b_2}^2x_1x_2 + 2ma_1a_2b_2{x_1}^2x_2 + a_2b_1b_2{x_1}^2x_2 + a_1{b_2}^2{x_1}^2x_2 + 2m^2{a_2}^2b_2{x_1}^3x_2 + 2ma_2{b_2}^2{x_1}^3x_2 + m{a_2}^2b_2{x_1}^2{x_2}^2 + a_2{b_2}^2{x_1}^2{x_2}^2) \bmod p$$

显然上式可以通过规定的检验。

第三步，建立联立方程组 T 并判断其是否有解：

针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，列出 **B**(***x***)中这些项的系数的多项式，并规定：令 **A**(***x***)中的每个多项式的关于 ***x*** 的已有的每一种项的类型，所对应的 **B**(***x***)中的这种项的系数的多项式（用 u_i 表示）都不为 0；令 **B**(***x***)中的每个多项式的相对于 **A**(***x***)而新产生的关于 ***x*** 的每一种项的类型，所对应的关于其系数的多项式（用 u'_i 表示）都为 0；从而列出联立方程组，即：

$$\mathbf{T}: \{u_1 \neq 0, u_2 \neq 0, \ldots, u'_1 = 0, u'_2 = 0, \ldots\}$$

具体对于上述实施例，先提取 z_1 中有关项的系数，令其为 0：

对于项 ${x_1}^3$：　　$ma_2(2ma_1a_2 + a_2b_1 + a_1b_2) = 0 \bmod p$

对于项 x_1^4： $m^2a_2^2(ma_2 + b_2) = 0 \bmod p$

对于项 $x_1^2 x_2$： $a_2(2ma_1a_2 + a_2b_1 + a_1b_2) = 0 \bmod p$

对于项 $x_1^3 x_2$： $2ma_2^2 (ma_2 + b_2) = 0 \bmod p$

对于项 $x_1^2 x_2^2$： $a_2^2 (ma_2 + b_2) = 0 \bmod p$

其它的项则令它们不等于 0；然后再提取 z_2 中有关项的系数，令其为 0：

对于项 x_1^3： $mb_2(2ma_1a_2 + a_2b_1 + a_1b_2) = 0 \bmod p$

对于项 x_1^4： $m^2a_2b_2(ma_2 + b_2) = 0 \bmod p$

对于项 $x_1^2 x_2$： $b_2(2ma_1a_2 + a_2b_1 + a_1b_2) = 0 \bmod p$

对于项 $x_1^3 x_2$： $2ma_2b_2 (ma_2 + b_2) = 0 \bmod p$

对于项 $x_1^2 x_2^2$： $a_2b_2 (ma_2 + b_2) = 0 \bmod p$

其它的项则令它们不等于 0。把上述方程组化简，则联立方程组 **T** 为：

$$\begin{cases} 2m\,a_1a_2 + a_2b_1 + a_1b_2 = 0 \mod p \\ m\,a_2 + b_2 = 0 \mod p \end{cases}$$

这是一个有限域 $\mathbf{F}_p$ 上的关于(a_1, a_2, b_1, b_2, m)的不定方程组，**T** 的通解为：

$$ma_1 + b_1 = 0 \bmod p，\ ma_2 + b_2 = 0 \bmod p$$

这说明上述的 $\mathbf{A}(\boldsymbol{x})$的表示形式可以设置为保形迭代变换。

通常情况下 **T** 为复杂的多变元非线性不定方程组，但建立该方程组的目的是求其任意一组特解，其难度比直接求不定方程组的通解容易。

第四步，求出方程组 T 的一组解，代入所期望的 $\mathbf{A}(\boldsymbol{x})$：

例如，对于上述实施例，设 p=17，a_0=1，b_0= 7，a_1=3，a_2= 5，m=2，则 $b_1 = -ma_1 \bmod 17 = 11$，$b_2 = -ma_2 \bmod 17 = 7$，可以证明这个 $\mathbf{A}(\boldsymbol{x})$是保形迭代函数：

$$\mathbf{A}(\boldsymbol{x}) = ((1 + 3x_1 + 10x_1^2 + 5x_1x_2) \bmod 17,$$
$$(7 + 11x_1 + 14x_1^2 + 7x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(2)}(\boldsymbol{x}) = ((15 + 8x_1 + 4x_1^2 + 2x_1x_2) \bmod 17,$$
$$(13 + x_1 + 9x_1^2 + 13x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(3)}(\boldsymbol{x}) = ((7 + 10x_1 + 5x_1^2 + 11x_1x_2) \bmod 17,$$
$$(12 + 14x_1 + 7x_1^2 + 12x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(4)}(\boldsymbol{x}) = ((14 + 4x_1 + 2x_1^2 + x_1x_2) \bmod 17,$$
$$(15 + 9x_1 + 13x_1^2 + 15x_1x_2) \bmod 17)$$

$$\mathbf{A}^{(5)}(\boldsymbol{x}) = ((10 + 5x_1 + 11x_1^2 + 14x_1x_2) \bmod 17,$$
$$(6 + 7x_1 + 12x_1^2 + 6x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(6)}(\boldsymbol{x}) = ((5 + 2x_1 + x_1^2 + 9x_1x_2) \bmod 17,$$
$$(16 + 13x_1 + 15x_1^2 + 16x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(7)}(\boldsymbol{x}) = ((3 + 11x_1 + 14x_1^2 + 7x_1x_2) \bmod 17,$$
$$(3 + 12x_1 + 6x_1^2 + 3x_1x_2) \bmod 17)$$
$$\mathbf{A}^{(8)}(\boldsymbol{x}) = ((9 + x_1 + 9x_1^2 + 13x_1x_2) \bmod 17,$$
$$(8 + 15x_1 + 16x_1^2 + 8x_1x_2) \bmod 17)$$

……

设 $\boldsymbol{x} = (3, 5)$，则 $\mathbf{A}(\boldsymbol{x}) = (5, 16)$，$\mathbf{A}^{(2)}(\boldsymbol{x}) = (3, 3)$，$\mathbf{A}^{(3)}(\boldsymbol{x}) = (9, 8)$，$\mathbf{A}^{(4)}(\boldsymbol{x}) = (8, 10)$，$\mathbf{A}^{(5)}(\boldsymbol{x}) = (11, 4)$，$\mathbf{A}^{(6)}(\boldsymbol{x}) = (2, 5)$，$\mathbf{A}^{(7)}(\boldsymbol{x}) = (12, 2)$，$\mathbf{A}^{(8)}(\boldsymbol{x}) = (16, 1)$，……。

上述的用 $\mathbf{F}_p$ 上的多项式来建立 $\mathbf{A}(\boldsymbol{x})$的方法，同样也适合于用 $\mathbf{F}_p$ 上的有理分式来建立 $\mathbf{A}(\boldsymbol{x})$，以及用整数剩余类环 $\mathbf{Z}_N$ 上的多项式来建立 $\mathbf{A}(\boldsymbol{x})$，并推广到 $n>2$ 的情形，只不过建立 $\mathbf{A}(\boldsymbol{x})$的推导过程更复杂。

需要注意的是：计算 $\mathbf{F}_p$ 上的有理分式的值时，会出现虽然分母不是其系数均为 0 的多项式、但分母多项式的值为 0 的情况，应采取必要的容错、纠错措施。

随着向量长度 n 的增加，$\mathbf{A}(\boldsymbol{x})$的函数规模将迅速增加，使得 $\mathbf{A}(\boldsymbol{x})$需要占用很大的存储空间。把一个很大的 $\mathbf{A}(\boldsymbol{x})$压缩成一个短数据是困难的。但是，可以把一个短数据$\mu_0$作为一个伪随机序列发生器的种子，用其产生的伪随机序列$(\mu_1, \mu_2, \ldots)$建立对应的 $\mathbf{A}(\boldsymbol{x})$，从而用短数据$\mu_0$表示对应的 $\mathbf{A}(\boldsymbol{x})$，更换 $\mathbf{A}(\boldsymbol{x})$只需要重新约定$\mu_0$即可。即优选的，可以依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；并采用该伪随机序列的种子，标识该 $\mathbf{A}(\boldsymbol{x})$。

具体过程介绍如下：

对于第一类 $\mathbf{A}(\boldsymbol{x})$，按照约定的规则，用$(\mu_1, \mu_2, \ldots)$确定 $\mathbf{A}(\boldsymbol{x})$中的每个系数，用$\mu_0$表示该 $\mathbf{A}(\boldsymbol{x})$；

对于第二类、第三类 $\mathbf{A}(\boldsymbol{x})$，按照约定的规则，用$(\mu_1, \mu_2, \ldots)$来确定 $\mathbf{A}(\boldsymbol{x})$：

首先，用$(\mu_1, \mu_2, \ldots)$来确定 $\mathbf{A}(\boldsymbol{x})$的函数形式，即确定 A(x)中的哪些项的系数不为 0，哪些项的系数为 0，例如在上述实施例中，

$$y_1 = \mathrm{A}_1(x_1, x_2) = (a_0 + a_1x_1 + ma_2x_1^2 + a_2x_1x_2) \bmod p$$
$$y_2 = \mathrm{A}_2(x_1, x_2) = (b_0 + b_1x_1 + mb_2x_1^2 + b_2x_1x_2) \bmod p$$

假定是用($\mu_1, \mu_2, \ldots$)的值，来确定其关于 x_2、x_2^2 的项的系数为 0，而 x_1、 x_1^2、x_1x_2 的项的系数不为 0；

其次，在确定了关于系数的方程组 **T** 有解后，用($\mu_1, \mu_2, \ldots$)的值，来确定该方程组的一组特解，例如在上述实施例中，用($\mu_1, \mu_2, \ldots$)的值，来确定 a_0=1，$b_0 = 7$，a_1=3，$a_2 = 5$，m=2，然后计算出 $b_1 = -ma_1 \bmod 17 = 11$，$b_2 = -ma_2 \bmod 17 = 7$，并用这些变量的值来确定 $\mathbf{A}(\boldsymbol{x})$的系数。

采用上述方法，在确定的伪随机序列($\mu_1, \mu_2, \ldots$)的控制下，一定能建立一个与该($\mu_1, \mu_2, \ldots$)相对应的 $\mathbf{A}(\boldsymbol{x})$，从而可以用一个短数据的$\mu_0$来表示一个长数据的 $\mathbf{A}(\boldsymbol{x})$。至于具体的对应过程，则由于其多种多样，所以在此仅仅以一个例子进行说明，其他方案就不再介绍了。

这种方法的突出优点是：可实现 $\mathbf{A}(\boldsymbol{x})$的高效率压缩编码，让不同的用户、根据不同的情况、使用不同的 $\mathbf{A}(\boldsymbol{x})$，从而做到了密码算法参数 $\mathbf{A}(\boldsymbol{x})$的勤换多变。实行这种技术体制，攻击者对于每种具体的 $\mathbf{A}(\boldsymbol{x})$，都要投入力量进行专门的密码分析，将大大增加破译的代价。

下面本发明提供了两种密钥协商方法的具体方式，其区别在于公开传递的信息是采用向量 $\boldsymbol{d}_i$，还是采用函数 $\mathbf{B}_i(\boldsymbol{x})$。

密钥协商方法 1

设密码参数 $\mathbf{A}(\boldsymbol{x}) = ((1+ 3x_1 + 10x_1^2 + 5x_1x_2) \bmod 17, (7 + 11x_1 + 14x_1^2 + 7x_1x_2) \bmod 17)$，$\boldsymbol{q}$=(3, 5)，执行：

用户 1 随机选择整数 k_1=2，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_1 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_1 = \mathbf{A}^{(k_1)}(\boldsymbol{q}) = \mathbf{A}^{(2)}(3, 5) = (3, 3)$，把计算结果 $\boldsymbol{d}_1$= (3, 3)传递给用户 2；

用户 2 随机选择整数 k_2=3，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_2 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_2 = \mathbf{A}^{(k_2)}(\boldsymbol{q}) = \mathbf{A}^{(3)}(3, 5) = (9, 8)$，把计算结果 $\boldsymbol{d}_2$= (9, 8)传递给用户 1；

用户 1 计算密钥 $\boldsymbol{K} = (K_1, \ldots, K_n) = \mathbf{A}^{(k_1)}(\boldsymbol{d}_2) = \mathbf{A}^{(2)}(9, 8) = (11, 4)$；

用户 2 计算密钥 $\boldsymbol{K} = (K_1, \ldots, K_n) = \mathbf{A}^{(k_2)}(\boldsymbol{d}_1) = \mathbf{A}^{(3)}(3, 3) = (11, 4)$；

从而，用户 1 和用户 2 建立了相同的密钥 $\boldsymbol{K}$= (11, 4)。

密钥协商方法 2

设密码参数 $\mathbf{A}(\boldsymbol{x}) = ((1+ 3x_1 + 10x_1^2 + 5x_1x_2) \bmod 17, (7 + 11x_1 + 14x_1^2 + 7x_1x_2) \bmod 17)$，执行：

用户 1 随机选择整数 k_1=2，推导出 k_1 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\mathbf{B}_1(\boldsymbol{x}) = \mathbf{A}^{(k_1)}(\boldsymbol{x}) = \mathbf{A}^{(2)}(\boldsymbol{x}) = ((15 + 8x_1 + 4x_1^2 + 2x_1x_2) \bmod 17, (13 + x_1 + 9x_1^2 + 13x_1x_2) \bmod 17)$，并把函数

组 $\mathbf{B}_1(\boldsymbol{x})$传递给用户 2；

用户 2 随机选择整数 k_2=3，推导出 k_2 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\mathbf{B}_2(\boldsymbol{x})=\mathbf{A}^{(k_2)}(\boldsymbol{x})=\mathbf{A}^{(3)}(\boldsymbol{x})$ =((7+ $10x_1+5x_1^2+11x_1x_2$) mod 17，(12 + $14x_1+7x_1^2+12x_1x_2$) mod 17)，并把函数组 $\mathbf{B}_2(\boldsymbol{x})$传递给用户 1；

用户 1 计算密钥 $\boldsymbol{K}=\mathbf{B}_2^{(k_1)}(\boldsymbol{x})=\mathbf{B}_2^{(2)}(\boldsymbol{x})=\mathbf{A}^{(2\times3)}(\boldsymbol{x})$ = ((5 + $2x_1+x_1^2+9x_1x_2$) mod 17，(16 + $13x_1+15x_1^2+16x_1x_2$) mod 17)；

用户 2 计算密钥 $\boldsymbol{K}=\mathbf{B}_1^{(k_2)}(\boldsymbol{x})=\mathbf{B}^{(3)}(\boldsymbol{x})=\mathbf{A}^{(3\times2)}(\boldsymbol{x})$ = ((5 + $2x_1+x_1^2+9x_1x_2$) mod 17，(16 + $13x_1+15x_1^2+16x_1x_2$) mod 17)；

从而，用户 1 和用户 2 建立了相同的密钥 $\boldsymbol{K}$。

参照图 4，公开了本发明一种用于编码和译码数字消息的方法，主要用于加解密情况，具体可以包括：

步骤 401，预置加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1,\ldots,y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1,\ldots,x_n),\ldots,\mathrm{A}_n(x_1,\ldots,x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x}))=\mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 402、选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

步骤 403、加密端选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，然后利用该中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

步骤 404、解密端利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，然后利用该中间密钥对加密结果进行解密。

上述实施例中，加密端将实时选择的整数 t 通过 t 的变换结果的方式传送给解密端，而解密端实际上在建立公钥的时候已经将私钥 k 的信息暗含在其中，因此，相当于双方交换了各自的信息 t 和 k，因此，可以很好的完成加密和解密。具体的关于 t 的变换规则，本发明并不需要加以限定，变换的目的在于防止第三方获得 t 的信息，并且解密端可以利用其得到中间密钥即可。当然，变换规则设定的好坏，可能影响到本发明在加密和解密过程中的安全性。

在本发明的另一个优选实施例中，如果还包括以下步骤：建立加密端和解密端共享的、变元数量大于 1 的向量 $\boldsymbol{q}$，公钥 $\boldsymbol{d}=(d_1,\ldots,d_n)=\mathbf{A}^{(k)}(\boldsymbol{q})$；则，

所述步骤 403 进一步包括：加密端选择整数 t，将公钥变换为关于 t 的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(t)}(\boldsymbol{d})$，然后利用该中间密钥 $\boldsymbol{K}$ 对明文 M 进行加密，$C=\mathrm{D}(M, \boldsymbol{K})$，传送包含加密结果 C 和 t 的变换结果 $\boldsymbol{v}$ 的密文 E 至解密端，$E=\{\boldsymbol{v}, C\}$，$\boldsymbol{v}=(v_1, \ldots, v_n)=\mathbf{A}^{(t)}(\boldsymbol{q})$ ；

所述步骤 404 进一步包括：解密端利用 t 的变换结果 $\boldsymbol{v}$、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(k)}(\boldsymbol{v})$，然后利用该中间密钥 $\boldsymbol{K}$ 对加密结果 C 进行解密，得到明文 M，$M=\mathrm{D}^{-1}(C, \boldsymbol{K})$；

其中，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k)}(\mathbf{A}^{(t)}(\boldsymbol{x}))=\mathbf{A}^{(k+t)}(\boldsymbol{x})$。

例如，设置共享的 $\boldsymbol{q}=(q_1, \ldots, q_n)$；选择整数 k；计算 $\boldsymbol{d}=(d_1, \ldots, d_n)=\mathbf{A}^{(k)}(\boldsymbol{q})$；把 k 作为私钥；把 $\boldsymbol{d}$ 作为公钥；

运用公钥 $\boldsymbol{d}$，把明文 M 转换成密文 E 的加密方法是：随机选择整数 t，计算：

$$\boldsymbol{v}=(v_1, \ldots, v_n)=\mathbf{A}^{(t)}(\boldsymbol{q}),\ \boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(t)}(\boldsymbol{d}),$$

$$C=\mathrm{D}(M, \boldsymbol{K}),\ E=\{\boldsymbol{v}, C\};$$

运用私钥 k，把密文 $E=\{\boldsymbol{v}, C\}$，转换成明文 M 的解密方法是：

$$\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(k)}(\boldsymbol{v}),\ M=\mathrm{D}^{-1}(C, \boldsymbol{K})。$$

上述的对称密码的加密变换“$C=\mathrm{D}(M, \boldsymbol{K})$”，以及对应的解密变换 “$M=\mathrm{D}^{-1}(C, \boldsymbol{K})$”的具体实现方法，均属于公知技术。

在本发明的另一个优选实施例中，当公钥 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$时，则，

所述步骤 403 进一步包括：加密端选择整数 t，将公钥变换为关于 t 的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=\mathbf{B}^{(t)}(\boldsymbol{x})$，然后利用该中间密钥 $\boldsymbol{K}$ 对明文 M 进行加密，$C=\mathrm{D}(M, \boldsymbol{K})$，传送包含加密结果 C 和 t 的变换结果 $\mathbf{V}(\boldsymbol{x})$的密文 E 至解密端，$E=\{\mathbf{V}(\boldsymbol{x}), C\}$，$\mathbf{V}(\boldsymbol{x})=\mathbf{A}^{(t)}(\boldsymbol{x})$；

所述步骤 404 进一步包括：解密端利用 t 的变换结果 $\mathbf{V}(\boldsymbol{x})$、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=\mathbf{V}^{(k)}(\boldsymbol{x})$，然后利用该中间密钥 $\boldsymbol{K}$ 对加密结果 C 进行解密，得到明文 M，$M=\mathrm{D}^{-1}(C, \boldsymbol{K})$；

其中，所述 A(x)进一步满足：若 B(x)=A(k)(x)，则 B(t)(x)=A(k t)(x)。

例如，选择整数 k；计算 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$；把 k 作为私钥；把 $\mathbf{B}(\boldsymbol{x})$作为公钥；

运用公钥 $\mathbf{B}(\boldsymbol{x})$，把明文 M 转换成密文 E 的加密方法是：随机选择整数 t，计算：

$$\mathbf{V}(\boldsymbol{x})=\mathbf{A}^{(t)}(\boldsymbol{x}),\ \boldsymbol{K}=\mathbf{B}^{(t)}(\boldsymbol{x}),$$

$C = \mathrm{D}(M, \boldsymbol{K})$，$E = \{\mathbf{V}(\boldsymbol{x}), C\}$；

运用私钥 k，把密文 $E = \{\mathbf{V}(\boldsymbol{x}), C\}$，转换成明文 M 的解密方法是：

$\boldsymbol{K} = \mathbf{V}^{(k)}(\boldsymbol{x})$，$M = \mathrm{D}^{-1}(C, \boldsymbol{K})$。

上述的对称密码的加密变换“$C = \mathrm{D}(M, \boldsymbol{K})$”，以及对应的解密变换“$M = \mathrm{D}^{-1}(C, \boldsymbol{K})$”的具体实现方法，均属于公知技术。

对于本部分关于加解密的实施例中 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$的建立过程，可以参见前述相关部分即可，在此不再详述。本实施例也可以依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；采用该伪随机序列的种子，标识该 $\mathbf{A}(\boldsymbol{x})$。

优选的，本实施例还可以建立基于身份的密钥管理体制，所谓基于身份（ID-based）的密钥管理，就是把用户的身份标识——诸如姓名、地址、电话等信息，直接作为用户的公钥。

例如，可以通过以下步骤建立私钥：预置 λ 个私钥表 $L_1, \ldots, L_\lambda$ 以及对应的公钥表 $\mathbf{G}_1, \ldots, \mathbf{G}_\lambda$，分布在 λ 个密钥分配中心；依据预置规则，根据用户的身份 ID 获得指向多个私钥表的指针；分别从所指向的多个私钥表中各获取一私钥分量，组合得到该用户的私钥。具体描述如下：

(1)、由 λ 个密钥分配中心各自独立地随机建立自己的足够大的私钥表 $L_1, \ldots, L_\lambda$ 以及对应的公钥表 $\mathbf{G}_1, \ldots, \mathbf{G}_\lambda$；其私钥表的每个记录的内容为一个正整数，用 L_{ij} 表示，i=1, … λ, j= 1, 2, …；其公钥表中的相关记录的内容为对应的公钥；把公钥表公开，把私钥表由各个密钥分配中心秘密保存；

(2)、设置一个单向函数，其输入是用户的 ID，其输出是指向 λ 个私钥表和 λ 个公钥表的 θ 个指针 $\eta_1, \eta_2, \ldots\eta_\theta$：$\{\eta_1, \eta_2, \ldots\eta_\theta\} = \mathrm{Hash(ID)}$；

(3)、身份标识为 ID 的用户的私钥为

$$k = \sum_{i=1}^{\lambda}\sum_{j=1}^{\theta} L_{i,\eta_j} ,$$

即每个授权用户，分别从 λ 个密钥分配中心各领取一私钥分量：

$$k(i) = \sum_{j=1}^{\theta} L_{i,\eta_j},\ i = 1,\ldots,\lambda ,$$

然后把这些私钥分量相加，合成为该授权用户的私钥：$k = k(1) + \ldots + k(\lambda)$；

(4)、当公钥采用向量“$\boldsymbol{d} = (d_1, \ldots, d_n) = \mathbf{A}^{(k)}(\boldsymbol{q})$”时，设置共享的 $\boldsymbol{q} = (q_1, \ldots, q_n)$，身份标识为 ID 的用户的公钥为：

$$\boldsymbol{d}=\mathbf{G}_{1,\eta_1}(\mathbf{G}_{1,\eta_2}(...(\mathbf{G}_{\lambda,\eta_{\theta-1}}(\mathbf{G}_{\lambda,\eta_\theta}(\boldsymbol{q})))...))=\mathbf{A}^{(\sum_{i=1}^{\lambda}\sum_{j=1}^{\theta}L_{i,\eta_j})}(\boldsymbol{q})\text{；}$$

当公钥采用函数“$\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$”时，身份标识为 ID 的用户的公钥为：

$$\mathbf{B}(\boldsymbol{x})=\mathbf{G}_{1,\eta_1}(\mathbf{G}_{1,\eta_2}(...(\mathbf{G}_{\lambda,\eta_{\theta-1}}(\mathbf{G}_{\lambda,\eta_\theta}(\boldsymbol{x})))...))=\mathbf{A}^{(\sum_{i=1}^{\lambda}\sum_{j=1}^{\theta}L_{i,\eta_j})}(\boldsymbol{x})\text{。}$$

本发明运用多个密钥分配中心联合建立用户私钥的方法，来实现基于身份的密钥管理体制，其特点是：用户的 ID 就是该用户的公钥；各个密钥分配中心、各个用户各自管理各自的秘密，谁也不能获得全部的秘密；各个密钥分配中心并不是由于行政管理制度和计算能力的制约、而是由于缺少信息，而无法窃取用户的私钥。

本发明提供两种加密方案，其区别在于公钥是采用向量 $\boldsymbol{d}$，还是采用函数 $\mathbf{B}(\boldsymbol{x})$；方案 1 用一个向量作为公钥，方案 2 用一个函数组作为公钥。加密方案 1 的优点是公钥的数据长度很短，加密方案 2 的优点是密码的安全性更强。下面分别具体说明：

加密方案 1

首先，设 n=2，设密码参数，即保形迭代变换函数为：

$\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, x_2)，\ \mathrm{A}_1(x_1, x_2))$，其中：

$$\mathrm{A}_1(x_1, x_2)=(1+3x_1+10{x_1}^2+5x_1x_2)\bmod 17，$$

$$\mathrm{A}_2(x_1, x_2)=(7+11x_1+14{x_1}^2+7x_1x_2)\bmod 17，$$

设置共享的向量 $\boldsymbol{q}=(q_1, q_2)=(3, 5)$，$q_i\in$有限域 $\mathbf{F}_p$，p=17，选择正整数 k=2 作为私钥，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$进行 k 层迭代，计算：

$$\boldsymbol{d}=(d_1, d_2)=\mathbf{A}^{(k)}(\boldsymbol{q})=\mathbf{A}^{(2)}(3, 5)=(3, 3)，$$

把 $\boldsymbol{d}=(3, 3)$作为公钥（其计算结果，见本说明书的如何建立第二类、第三类 $\mathbf{A}(\boldsymbol{x})$的小数据实施例）；

运用公钥 $\boldsymbol{d}$ 加密时，随机选择一个正整数 t，设 t=3，把 t 和 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$，计算：

$$\boldsymbol{v}=\mathbf{A}^{(t)}(\boldsymbol{q})=\mathbf{A}^{(3)}(3, 5)=(9, 8)，$$

把 t 和公钥 $\boldsymbol{d}$ 代入 $\mathbf{A}(\boldsymbol{x})$，计算：

$$\boldsymbol{K}=\mathbf{A}^{(t)}(\boldsymbol{d})=\mathbf{A}^{(3)}(3, 3)=(11, 4)，$$

把 $\boldsymbol{K}$ 作为对称加密使用的密钥，进行加密变换：

$$C = \mathrm{D}(M, \boldsymbol{K}) = \mathrm{D}(M, (11, 4)),$$

其中，“$C = \mathrm{D}(M, \boldsymbol{K})$”可选用任意一种对称密码加密算法，例如，采用美国数据加密标准 AES；以下将用“$M = \mathrm{D}^{-1}(C, \boldsymbol{K})$”表示与对应的对称密码解密算法；

上述的用公钥 $\boldsymbol{d}$ 加密的结果由两部分组成：$\{\boldsymbol{v}, C\}=\{(9, 8), C\}$，其中 $\boldsymbol{v}=(9, 8)$是密文报头，C 是密文正文；

运用私钥 k 解密时，先把密文头 $\boldsymbol{v}$ 和私钥 k 代入 $\mathbf{A}(\boldsymbol{x})$，计算：

$$\boldsymbol{K} = \mathbf{A}^{(k)}(\boldsymbol{v}) = \mathbf{A}^{(2)}(9, 8) = (11, 4),$$

把 $\boldsymbol{K}$ 作为对称解密使用的密钥，进行解密变换：

$$M = \mathrm{D}^{-1}(C, \boldsymbol{K}) = \mathrm{D}(M, (11, 4))$$

由于加密和解密使用了相同的 $\boldsymbol{K}=(11, 4)$，因此一定可以恢复出正确的明文。

加密方案 2

两种加密方案的区别仅仅在于其公钥的数据格式不同：加密方案 1 用一个向量 $\boldsymbol{d}=(d_1, \ldots, d_n)$作为公钥，其优点是公钥数据长度短；加密方案 2 则用一个函数组 $\mathbf{B}(\boldsymbol{x})= (\mathrm{B}_1(x_1, \ldots, x_n), \ldots, \mathrm{B}_n(x_1, \ldots, x_n))$作为公钥，其优点是可以获得更长的密码周期。

设 $n=2$，$p=17$，设密码参数：

$$\mathbf{A}(\boldsymbol{x}) = (\mathrm{A}_1(x_1, x_2),\ \mathrm{A}_1(x_1, x_2))$$

$$= ((1+ 3x_1 + 10{x_1}^2 + 5x_1x_2) \bmod 17,\ (7 + 11x_1 + 14{x_1}^2 + 7x_1x_2) \bmod 17),$$

设私钥 $k=2$，其对应的公钥为：

$$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}^{(k)}(\boldsymbol{x}) = \mathbf{A}^{(2)}(\boldsymbol{x})$$

$$= ((15 + 8x_1 + 4{x_1}^2 + 2x_1x_2) \bmod 17,\ (13 + x_1 + 9{x_1}^2 + 13x_1x_2) \bmod 17);$$

运用公钥 $\mathbf{B}(\boldsymbol{x})$加密时，随机选择一个正整数 $t=3$，计算密文报头：

$$\mathbf{V}(\boldsymbol{x}) = \mathbf{A}^{(t)}(\boldsymbol{x}) = \mathbf{A}^{(3)}(\boldsymbol{x})$$

$$= ((7+ 10x_1 + 5{x_1}^2 + 11x_1x_2) \bmod 17,\ (12 + 14x_1 + 7{x_1}^2 + 12x_1x_2) \bmod 17),$$

计算出对称加密使用的密钥：

$$\boldsymbol{K} = \mathbf{B}^{(t)}(\boldsymbol{x}) = \mathbf{B}^{(3)}(\boldsymbol{x}) = \mathbf{A}^{(3\times 2)}(\boldsymbol{x}) = \mathbf{A}^{(6)}(\boldsymbol{x})$$

$$= ((5 + 2x_1 + {x_1}^2 + 9x_1x_2) \bmod 17,\ (16 + 13x_1 + 15{x_1}^2 + 16x_1x_2) \bmod 17),$$

运用 $\boldsymbol{K}$ 进行加密变换，得到密文正文为：

$$C = \mathrm{D}(M, \boldsymbol{K})$$

$= \mathrm{D}(M, ((5+2x_1+x_1^2+9x_1x_2) \bmod 17, (16+13x_1+5x_1^2+16x_1x_2) \bmod 17))$，

上述的用公钥 $\mathbf{B}(\boldsymbol{x})$加密的结果由两部分组成：

$$\mathrm{E} = \{\mathbf{V}(\boldsymbol{x}), C\}$$

$=\{((7+10x_1+5x_1^2+11x_1x_2) \bmod 17, (12+14x_1+7x_1^2+12x_1x_2) \bmod 17), \quad C\}$；

运用私钥 k 解密时，先把私钥 k 和代入密文头 $\mathbf{V}(\boldsymbol{x})$，计算出对称加密使用的密钥：

$$\boldsymbol{K} = \mathbf{V}^{(k)}(\boldsymbol{x}) = \mathbf{V}^{(2)}(\boldsymbol{x}) = \mathbf{A}^{(2\times 3)}(\boldsymbol{x})$$

$= ((5 + 2x_1 + x_1^2 + 9x_1x_2) \bmod 17, (16 + 13x_1 + 15x_1^2 + 16x_1x_2) \bmod 17)$，

然后运用 $\boldsymbol{K}$ 把密文正文 C 转换为明文 M：

$$M = \mathrm{D}^{-1}(C, \boldsymbol{K})$$

$= \mathrm{D}(M, ((5+2x_1+x_1^2+9x_1x_2) \bmod 17, (16+13x_1+15x_1^2+16x_1x_2) \bmod 17))$，

由于加密和解密使用了相同的 $\boldsymbol{K}$，因此可以恢复出正确的明文。

参照图5，示出了一种用于数字签名及验证的方法实施例，包括：

步骤 501，建立签名端和验证端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 502、选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

步骤 503、签名端选择整数 t，依据预置规则将待签名数据变换为与 t、私钥 k 相关的中间消息，然后传送包含中间消息和 t 的变换结果的数字签名至验证端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

步骤 504、验证端利用 t 的变换结果、待签名数据、中间消息、公钥和 $\mathbf{A}(\boldsymbol{x})$ 验证是否满足所述预置规则，如果满足，则该数字签名验证通过。

由于预置规则是双方私下确定的，所以可以保证签名的有效性。至于具体的规则，本发明无法也无需加以限定，本领域技术人员根据需要自行设定即可。

一般情况下，直接验证是否满足所述预置规则；优选的，也可以对该预置规则进行变换，通过验证是否满足变换后的预制规则来验证签名是否正确。

在本发明的一个优选实施例中，还包括：建立签名端和验证端共享的、变元数量大于 1 的向量 $\boldsymbol{q}$；则，

所述步骤 503 进一步包括：签名端选择整数 t，依据预置规则将待签名数据 M 变换为与 t、私钥 k 相关的中间消息 c，然后传送包含中间消息 c 和 t 的变换结果 $\boldsymbol{e}$ 的数字签名 S 至解密端，$S=\{c, \boldsymbol{e}\}$；所述 t 的变换结果 $\boldsymbol{e}$ 与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关：$\boldsymbol{e}=(e_1, \ldots, e_n)=\mathbf{A}^{(t)}(\boldsymbol{q})$；其中，所述预置规则为整数方程 Φ：$c=\Phi(t, w, k)$，$w$ 为依据待签名数据 M 计算得到的整数；

所述步骤 504 进一步包括：验证端利用 t 的变换结果 e、依据待签名数据 M 计算得到的 w、中间消息 c、公钥和 $\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则：假设整数方程 Φ 可进一步表示为: $\alpha=\beta$，并且β中包含 t，则验证 $\mathbf{A}^{(\alpha)}(\boldsymbol{q})=\mathbf{A}^{(\beta)}(\boldsymbol{q})=\mathbf{A}^{(\beta-t)}(\boldsymbol{e})$ 是否成立；如果正确，则该数字签名验证通过；

其中，当公钥 $\boldsymbol{d}=(d_1, \ldots, d_n)=\mathbf{A}^{(k)}(\boldsymbol{q})$时，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k)}(\mathbf{A}^{(t)}(\boldsymbol{x}))=\mathbf{A}^{(k+t)}(\boldsymbol{x})$；当公钥 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$时，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$，则 $\mathbf{B}^{(t)}(\boldsymbol{x})=\mathbf{A}^{(kt)}(\boldsymbol{x})$。

对于本部分关于数字签名的实施例中 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$的建立过程，可以参见前述相关部分即可，在此不再详述。另外，本实施例也可以依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；采用该伪随机序列的种子，标识该 $\mathbf{A}(\boldsymbol{x})$。以及，本实施例也可以适用于构建基于身份的密钥管理体制的情况，这些都在前面已经详述，所以在此不再赘述。

本发明还提供了两种数字签名方案，其区别在于公钥是采用向量$\boldsymbol{d}$，还是采用函数$\mathbf{B}(\boldsymbol{x})$，具体说明如下：

数字签名方案 1

如图 6、7 所示：设 $w=\mathrm{H}(\boldsymbol{\Delta})$是单向函数，该函数的输入 $\boldsymbol{\Delta}$ 为验证方可以获得的诸如数据 M 等信息的某种组合，但至少应包括 M，其输出 w 为正整数；

设 Φ 是关于 c、t、w、k 的整数方程，该方程可采用不同的形式，例如：$k=c+w+t$，$k+w=c+t$，$k+c+w=t$，…；即把 c、t、w、k 划分成均没有使用减号的α和β两部分，并且β中包含了 t，从而把方程 Φ，以及对应的迭代方程 Φ’表示为：

$$\Phi: \alpha=\beta,$$

$$\Phi': \mathbf{A}^{(\alpha)}(\boldsymbol{x})=\mathbf{A}^{(\beta)}(\boldsymbol{x});$$

记“$c=\Phi(t, w, k)$”是把已知的 t、w、k 代入方程 Φ 求 c 的运算。不同的 Φ 的计算速度有差别，但安全性相同。例如在本实施例中，规定

$$\Phi: k=c+w+t,$$

$$\Phi'：\ \mathbf{A}^{(k)}(\boldsymbol{x})=\mathbf{A}^{(c+w+t)}(\boldsymbol{x});$$

把 $\boldsymbol{d}=\mathbf{A}^{(k)}(\boldsymbol{q})$、$\boldsymbol{e}=\mathbf{A}^{(t)}(\boldsymbol{q})$代入上式后，其具体的检验公式为

$$\Phi'：\ \boldsymbol{d}=\mathbf{A}^{(c+w)}(\boldsymbol{e})?$$

规定了整数方程 Φ 以后，为保证 Φ 一定有解，还需要为 c、t、w、k 各规定一个范围。例如当 $c=\Phi(t, w, k)=k-w-t$ 时，要求：$k>c$，$k>w+t$。

如图 6、7 所示：设密码参数 $\mathbf{A}(\boldsymbol{x})=((1+3x_1+10x_1^2+5x_1x_2) \bmod 17, (7+11x_1+14x_1^2+7x_1x_2) \bmod 17)$；设 $\boldsymbol{q}=(3, 5)$，私钥 $k=8$，其对应的公钥 $\boldsymbol{d}=\mathbf{A}^{(k)}(\boldsymbol{q})=\mathbf{A}^{(8)}(3, 5)=(16, 1)$；

运用私钥 k，把数据 M，转换成关于 M 的数字签名 S 的方法是：随机选择一个正整数 t，设 $t=3$，计算：

$$\boldsymbol{e}=(e_1, e_2)=\mathbf{A}^{(t)}(\boldsymbol{q})=\mathbf{A}^{(3)}(3, 5)=(9, 8),$$

假设 $w=\mathrm{H}(\boldsymbol{\Delta})=\mathrm{H}(M)=2$，则 $c=\Phi(t, w, k)=k-w-t=8-2-3=3$，其签名为：$S=\{c, \boldsymbol{e}\}=\{3, (9, 8)\}$；

运用公钥 $\boldsymbol{d}=(16, 1)$，检验关于 M 的数字签名 S 是否正确的方法是：先计算 $w=\mathrm{H}(\boldsymbol{\Delta})=\mathrm{H}(M)=2$，然后把 $\boldsymbol{d}$、c、w、$\boldsymbol{e}$、$\boldsymbol{q}$ 代入迭代方程

$$\Phi'：\ \mathbf{A}^{(k)}(\boldsymbol{x})=\mathbf{A}^{(c+w+t)}(\boldsymbol{x}),$$

由于 $\boldsymbol{d}=\mathbf{A}^{(k)}(\boldsymbol{q})$，$\boldsymbol{e}=\mathbf{A}^{(t)}(\boldsymbol{q})$，则具体的验证公式为

$$\boldsymbol{d}=\mathbf{A}^{(c+w)}(\boldsymbol{e})\ ?$$

$$(16, 1)=\mathbf{A}^{(3+2)}(9, 8)=\mathbf{A}^{(5)}(9, 8)$$

因此 $S=\{3, (9, 8)\}$表示 M 的签名得到了验证。

数字签名方案 2

如图 6、7 所示：设 $w=\mathrm{H}(\boldsymbol{\Delta})$是单向函数，该函数的输入 $\boldsymbol{\Delta}$ 为验证方可以获得的诸如数据 M 等信息的某种组合，但至少应包括 M；其输出 w 为正整数；

设 Φ 是关于 c、t、w、k 的整数方程，该方程可采用不同的形式，例如：$wk=c+t$，$wk+c=t$，…；即把 c、t、w、k 划分成均没有使用减号的α和β两部分，方程中可以有包含了 k 的两个变量的乘积项（如 wk），并且β中包含了 t，从而把方程 Φ，以及对应的迭代方程 Φ’表示为：

$$\Phi:\ \alpha=\beta,$$

$$\Phi_1':\ \mathbf{A}^{(\alpha)}(\boldsymbol{x})=\mathbf{A}^{(\beta)}(\boldsymbol{x});$$

记“$c=\Phi(t, w, k)$”是把已知的 t、w、k 代入方程 Φ 求 c 的运算。不同的 Φ 的计算速度有差别，但安全性相同。例如在本实施例中，规定

$$\Phi：\ wk = c + t,$$

$$\Phi'：\ \mathbf{A}^{(wk)}(\boldsymbol{x}) = \mathbf{A}^{(c+t)}(\boldsymbol{x});$$

把 $\mathbf{B}(\boldsymbol{q})=\mathbf{A}^{(k)}(\boldsymbol{q})$、$\boldsymbol{e}=\mathbf{A}^{(t)}(\boldsymbol{q})$代入上式后，具体的验证公式为

$$\mathbf{B}^{(w)}(\boldsymbol{q}) = \mathbf{A}^{(c)}(\boldsymbol{e})?$$

规定了整数方程 Φ 以后，为保证 Φ 一定有解，还需要为 c、t、w、k 各规定一个范围。例如，当 $c = \Phi(t, w, k) = wk–t$ 时，要求：$wk > c$。

设密码参数 $\mathbf{A}(\boldsymbol{x}) = ((1+ 3x_1 + 10x_1^2 + 5x_1x_2) \bmod 17, (7 + 11x_1 + 14x_1^2 + 7x_1x_2) \bmod 17)$；设 $\boldsymbol{q} = (3, 5)$，私钥 $k = 2$，其对应的公钥为 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}^{(k)}(\boldsymbol{x}) = \mathbf{A}^{(2)}(\boldsymbol{x}) = ((15 + 8x_1 + 4x_1^2 + 2x_1x_2) \bmod 17, (13 + x_1 + 9x_1^2 + 13x_1x_2) \bmod 17)$；

运用私钥 k，把数据 M，转换成关于 M 的数字签名 S 的方法是：随机选择一个正整数 t，例如设 t=3，计算：

$$\boldsymbol{e} = (e_1, e_2) = \mathbf{A}^{(t)}(\boldsymbol{q}) = \mathbf{A}^{(3)}(3, 5) = (9, 8),$$

假设 $w = \mathrm{H}(\boldsymbol{\Delta}) = \mathrm{H}(M) = 4$，则 $c = \Phi(t, w, k) = wk - t = 4\times2 - 3 = 5$，其签名为：$S = \{c, \boldsymbol{e}\} = \{5, (9, 8)\}$；

运用公钥 $\mathbf{B}(\boldsymbol{x})$，检验关于 M 的数字签名 S 是否正确的方法是：先计算 $w = \mathrm{H}(\boldsymbol{\Delta}) = \mathrm{H}(M) = 4$，然后把 $\mathbf{B}(\boldsymbol{x})$、$\boldsymbol{q}$、$c$、$w$、$\boldsymbol{e}$ 代入迭代方程

$$\Phi'：\ \mathbf{B}^{(w)}(\boldsymbol{q}) = \mathbf{A}^{(c)}(\boldsymbol{e})\ ?$$

其中：

$$\mathbf{B}^{(w)}(\boldsymbol{q}) = \mathbf{B}^{(4)}(3, 5) = \mathbf{A}^{(4\times2)}(3, 5) = \mathbf{A}^{(8)}(3, 5) = (16, 1)$$

$$\mathbf{A}^{(c)}(\boldsymbol{e}) = \mathbf{A}^{(5)}(9, 8) = (16, 1)$$

因此 $S = \{5, (9, 8)\}$作为对 M 的签名得到了验证。

数字签名方案的可扩展性说明

一旦建立 $\mathbf{A}(\boldsymbol{x})$后，本领域的技术人员通过对上述数字签名方案的理解和启迪，一定能设计出许多种看起来原理更复杂，编码技巧却十分相似的新的数字签名方案。例如，可以设置更复杂的方程 Φ，单向函数 H(.)的使用方式可以更灵活，当 $\mathbf{A}(\boldsymbol{x})$可逆时 k 还可以用负整数。建立 Φ、H(.)的具体方法属于公知技术（详见《应用密码学——协议、算法与 C 程序》，Bruce Schneier，机械工业出版社（China Machine Press），2000., pp.389-399）。然而，这些修改的数字签名方案都将遵循共同的本发明的必要技术特征：**其安全性基于多变元非线性保形迭代变换的层数问题**。

对于前述的各方法实施例，为了简单描述，故将其都表述为一系列的动作

组合，但是本领域技术人员应该知悉，本发明并不受所描述的动作顺序的限制，因为依据本发明，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定是本发明所必须的。

相应的，本发明还公开了一种密钥协商的系统实施例，具体包括：

共享单元，用于存储用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

中间结果交换单元，连接所述用户群中的各用户端，用于传递所述用户群中的各用户与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果至其他用户；

密钥计算单元，位于所述用户群中的各用户端，用于针对各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

相应的，本发明还公开了一种用于编码和译码数字消息的系统，包括：

共享单元，用于存储加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

加密单元，位于加密端，用于选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，然后利用该中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

解密单元，位于解密端，用于利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，然后利用该中间密钥对加密结果进行解密。

相应的，本发明还公开了一种用于数字签名及验证的系统，包括：

共享单元，用于存储签名端和验证端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向

量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

签名单元，位于签名端，用于选择整数 t，依据预置规则将待签名数据变换为与 t、私钥 k 相关的中间消息，然后传送包含中间消息和 t 的变换结果的数字签名至验证端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

验证单元，位于验证端，用于利用t的变换结果、待签名数据、中间消息、公钥和$\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则，如果满足，则该数字签名验证通过。

对于装置实施例而言，由于其基本相应于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。并且，在本发明的装置实施例中，其相应的模块单元都是针对相应的执行步骤虚拟出来的，为了节约篇幅，在此就不针对前述的各个流程步骤一一对应描述了，但是本领域技术人员应该知悉，各个执行步骤都是可以一一对应虚拟模块的。下面以一个例子进行简单说明：

前述的装置实施例都还可以包括一$\mathbf{A}(\boldsymbol{x})$建立单元，具体包括以下模块：

$\mathbf{A}(\boldsymbol{x})$结构确定模块，用于预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个环 $\mathbf{R}$ 上的 n 元多项式组成：其含有关于$(x_1, \ldots, x_n)$的大于 1 次的项；

参数确定模块，用于接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n、变元的数据长度和最高的非线性次数；

随机生成模块，用于依据所述指标参数和预置结构，生成一个 $\mathbf{A}(\boldsymbol{x})$的表示形式，该 $\mathbf{A}(\boldsymbol{x})$中的不为零的系数用变元符号表示；

迭代模块，用于将 $\mathbf{A}(\boldsymbol{x})$代入自身并执行展开、化简的数据处理：$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$；

方程组建立模块，用于针对 $\mathbf{B}(\boldsymbol{x})$与 $\mathbf{A}(\boldsymbol{x})$对比而新出现的每一个关于 $\boldsymbol{x}$ 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

判断模块，用于判断该方程组是否有解，如果没有解，则返回随机生成模

块；如果有解，则计算得到该方程组的一组解，并作为 **A(*x*)**中的系数的值，代入随机生成模块生成的 **A(*x*)**的表示形式；

结果输出模块，用于输出所得到的 **A(*x*)**。

本说明书中的各个实施例均基于同一技术构思，所以在描述时重点说明的都是该实施例的独特之处，各个实施例之间相同相似的部分互相参见即可。

由于本发明技术复杂、核心构思比较抽象，为了方便理解，下面对本发明和现有技术的主要区别进行简单描述：

现有技术中与本发明最接近的技术解决方案是 DH，即 1976 年由 Diffie 和 Hellman 提出的密钥协商协议，以及 ElGamal 提出的加密与数字签名方案等。其安全性基于：已知素数 p、本原元 $1<g<p$、以及 $d=g^k \bmod p$，求离散对数 k 是困难的。

Diffie-Hellman密钥协商协议为：两个用户运用共同约定的p、g，在公开信道上建立双方共同秘密约定的密钥K时，执行：

第一步：用户 1 随机选择正整数 k_1，计算 $d_1=g^{k_1} \bmod p$，并传递给用户 2；

第二步：用户 2 随机选择正整数 k_2，计算 $d_2=g^{k_2} \bmod p$，并传递给用户 1；

第三步：用户 1 计算密钥 $K=d_2^{k_1}=g^{k_2 k_1} \bmod p$；

第四步：用户 2 计算密钥 $K=d_1^{k_2}=g^{k_1 k_2} \bmod p$。

ElGamal加密方案为：运用共同约定的p、g，随机选择正整数k，公钥为$d=g^k \bmod p$，私钥为k；

加密算法为：随机选择正整数t，t与$(p-1)$互素，计算$a=g^t \bmod p$，$b=d^t M \bmod p$，其中M为明文，a、b为密文；

解密算法为：$M=b/a^k \pmod p$；

其中，“$d^t M$”和“b/a^k”可理解为简单的对称密码加解密运算。

ElGamal数字签名方案为：运用共同约定的p、g，随机选择正整数k，公钥为$d=g^k \bmod p$，私钥为k；

签名算法为：随机选择正整数t，t与$(p-1)$互素，计算：$a=g^t \bmod p$，b满足 $M=(ka+tb) \bmod (p-1)$，把a、b作为签名；

验证算法为：如果$d^a a^b \bmod p=g^M \bmod p$，则该签名得到验证；

其中，签名方程“$M=(ka+tb) \bmod (p-1)$”，以及对应的验证方程“$d^a a^b \bmod$

$p = g^M \bmod p$”，也可采用不同的形式。一般的签名方程可表示为：

$$\alpha t = (\beta + \omega k) \bmod (p - 1),$$

根据α、β、ω的±，以及取值是否为1，可以建立不同的签名方程（详见《应用密码学——协议、算法与C程序》，Bruce Schneier，机械工业出版社（China Machine Press），2000. , pp.389-399）。

当年的 DH 作为一个开拓性的发明，首次证明了“在完全公开的信道上，即使通信双方没有任何事先共同约定的秘密，也能进行保密通信”，这是密码学几千年来最有革命性的进展，它的贡献主要是提出了新概念。但 DH 的具体算法的安全性还存在很大的上升空间。

本发明与 DH 的主要区别在于：两者的安全性所基于的数学难题不同。

DH 的安全性基于有限域上的离散对数问题：已知素数 p、本原元 $1<g<p$、以及 $d = g^k \bmod p$，求离散对数 k 是困难的。各种 DH 密码算法的共同的必要技术特征是：以某个整数 k 为秘密参数，以 g 的 k 次幂 $d\,(= g^k \bmod p)$为公开参数，则由公开参数 d 求秘密参数 k 是困难的。其中，密钥协商协议中的公开参数、秘密参数的功能，等价于加密或签名时使用的公钥和私钥。

本发明的安全性基于多变元非线性保形迭代变换的迭代层数问题，即：设 $\mathbf{A}(\boldsymbol{x})$为给定的非线性保形迭代函数组，$k$ 为正整数，$\mathbf{B}(\boldsymbol{x})$为 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代，则已知 $\mathbf{A}(\boldsymbol{x})$、$\mathbf{B}(\boldsymbol{x})$求 k 是困难的。其中，$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}^{(k)}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\ldots(\mathbf{A}(\mathbf{A}(\boldsymbol{x})))\ldots))$，可理解为把 k 个 n 输入、n 输出的变换 $\mathbf{A}(\boldsymbol{x})$串联起来，合成为一个 n 输入、n 输出的变换 $\mathbf{B}(\boldsymbol{x})$，如图 8 所示，其中，虚框 801 表示 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代。

本发明的各种算法的核心构思在于：预设 $\mathbf{A}(\boldsymbol{x})$，以某个整数 k 为秘密参数，以 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代 $\mathbf{B}(\boldsymbol{x})$为公开参数，则由公开参数 $\mathbf{B}(\boldsymbol{x})$求秘密参数 k 是困难的，由秘密参数 k 求公开参数 $\mathbf{B}(\boldsymbol{x})$是容易的。

上述的难题还可采用另一种等价的表示：设 $\boldsymbol{q}=(q_1, \ldots, q_n)$，$\boldsymbol{d} = (d_1, \ldots, d_n) = \mathbf{A}^{(k)}(\boldsymbol{q}) = \mathbf{B}(\boldsymbol{x})$，则已知 $\boldsymbol{d}$、$\boldsymbol{q}$、$\mathbf{A}(\boldsymbol{x})$求 k 是困难的。也就是说，用从 $\boldsymbol{q}$ 到 $\boldsymbol{d}$ 的向量的数值的变化，来替代导致这些向量发生变化的从 $\mathbf{A}(\boldsymbol{x})$到 $\mathbf{B}(\boldsymbol{x})$的函数的变化，如图 9 所示，其中，虚框 901 表示 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代。

即，本发明的各种算法的核心构思的另一种表述为：预设 $\boldsymbol{q}$、$\mathbf{A}(\boldsymbol{x})$，以正整数 k 为秘密参数，以向量 $\boldsymbol{d}$ 为公开参数，则由公开参数 $\boldsymbol{d}$ 求秘密参数 k 是困

难的，由秘密参数 k 求公开参数 $\boldsymbol{d}$ 是容易的。虽然没有直接把具体的 $\mathbf{B}(\boldsymbol{x})$表达式作为公开参数，但是在由 k 计算 $\boldsymbol{d}$ 的过程中，需要使用 $\mathbf{B}(\boldsymbol{x})$。这种表述的优点是：$\boldsymbol{d}$ 的数据长度比 $\mathbf{B}(\boldsymbol{x})$的函数长度明显减少，节约了公钥的存储空间。

本发明与 DH 相比，使密码的安全性获得了显著提高，其有益效果体现在：与 DH 相比，本发明运行在一个更大、更复杂的算法空间中，使密码函数的规模发生爆炸。

在 DH 方案中，无论 k 多么大，其密码函数始终为关于变元 g 的一个单项式：

$$\text{当 } k=1 \text{ 时，} d = g \bmod p;$$

$$\text{当 } k=2 \text{ 时，} d = g^2 \bmod p;$$

$$\cdots\cdots;$$

$$\text{当 } k=s \text{ 时，} d = g^s \bmod p;$$

其数学性质非常简单、清晰，容易给密码破译提供线索。例如，利用其密码周期为 $p-1$ 的性质，可以用 Shor 量子算法进行破译（详见 P. W. Shor, “Algorithms for quantum computation: Discrete log and factoring ”, Proceedings of the 35th Symposium on Foundations of Computer Science, 1994, pp.124-134.）。

与 DH 相比，本发明经过迭代后，一方面，其关于变元 $\boldsymbol{x}$ 的次数保持不变；另一方面，不仅其关于系数的非线性次数增加，而且其关于系数的函数规模也发生了爆炸，从而大大提高了进行数学分析的困难性。例如，看起来非常简单的只有两个变元(x_1, x_2)的第一类 $\mathbf{A}(\boldsymbol{x})$，在经过 k 层迭代后的密码函数为：

当 k=1 时，

$$\mathbf{A}^{(1)}(\boldsymbol{x}) = \left(\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2} \bmod p,\quad \frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2} \bmod p\right);$$

当 k=2 时，

$$\mathbf{A}^{(2)}(\boldsymbol{x}) = \left(\frac{a_{10}+a_{11}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{12}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}} \bmod p,\right.$$

$$\left.\frac{a_{20}+a_{21}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{22}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}} \bmod p\right);$$

当 k=3 时，

$$\mathbf{A}^{(3)}(\boldsymbol{x})=$$

$$=\left(\frac{a_{10}+a_{11}\dfrac{a_{10}+a_{11}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{12}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}+a_{12}\dfrac{a_{20}+a_{21}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{22}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}}{b_0+b_1\dfrac{a_{10}+a_{11}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{12}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}+b_2\dfrac{a_{20}+a_{21}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{22}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}}\bmod p,\right.$$

$$\left.\frac{a_{20}+a_{21}\dfrac{a_{10}+a_{11}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{12}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}+a_{22}\dfrac{a_{20}+a_{21}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{22}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}}{b_0+b_1\dfrac{a_{10}+a_{11}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{12}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}+b_2\dfrac{a_{20}+a_{21}\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+a_{22}\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}{b_0+b_1\frac{a_{10}+a_{11}x_1+a_{12}x_2}{b_0+b_1x_1+b_2x_2}+b_2\frac{a_{20}+a_{21}x_1+a_{22}x_2}{b_0+b_1x_1+b_2x_2}}}\bmod p\right);$$

……；

把 $\mathbf{A}(\boldsymbol{x})$中的系数 a_{00}, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, b_0, b_1, b_2 看作为 $\mathbf{A}(\boldsymbol{x})$中的 9 个变量，也就是把 $\mathbf{A}(\boldsymbol{x})$理解为：

$$\boldsymbol{y}=(y_1, \ldots, y_n)=\mathbf{A}(\boldsymbol{x}, a_{00}, \ldots, b_2)$$
$$=(\mathrm{A}_1(x_1, \ldots, x_n, a_{00}, \ldots, b_2), \ldots, \mathrm{A}_n(x_1, \ldots, x_n, a_{00}, \ldots, b_2)),$$

则经过展开、化简后的 $\mathbf{A}^{(k)}(\boldsymbol{x})$中的分子多项式或分母多项式中关于系数($a_{00}$, …, b_2)的项的数量大约为：

$$\mathrm{C}_{9+k}^{k}=\frac{(9+k)!}{9!k!},$$

即 $\mathbf{A}(\boldsymbol{x}, a_{00}, \ldots, b_2)$由关于($a_{00}$, …, b_2)的稠密多项式组成。例如当 k 仅为 65536（即 16 比特二进制数 1000000000000000）时，其每个多项式的项的数量将大约增加到：614970856015462823268936355508844880385；

规模如此巨大的多项式，虽然在数学世界中是客观存在的，但需要占用指数级的存储空间，实际上是难以操作的，通过数学分析来探索其中的规律就更加困难。

按照当代计算数学的进展，运用数学分析来进行破译（例如对有 614970856015462823268936355508844880385 个项的函数进行分析）的代价，将远远大于直接用穷举法来破译（例如只进行 65536 次试验）的代价。

与上述情况相比，当采用第二类、第三类 $\mathbf{A}(\boldsymbol{x})$时，随着 k 的增加，$\mathbf{A}^{(k)}(\boldsymbol{x})$

的关于系数的非线性次数将以更快的速度增加，从而使 $\mathbf{A}^{(k)}(\boldsymbol{x})$关于系数的函数规模以更快的速度发生爆炸。例如，当 $\mathbf{A}(\boldsymbol{x})$中含有 x_1x_2 的项时，$\mathbf{A}^{(k)}(\boldsymbol{x})$关于系数的非线性次数为$(2^k-1)$。

运用 Shor 量子算法对本发明进行破译时，需要对 $\mathbf{A}^{(k)}(\boldsymbol{x})$的函数序列$\{\mathbf{A}^{(1)}(\boldsymbol{x}), \mathbf{A}^{(2)}(\boldsymbol{x}), \ldots, \mathbf{A}^{(s)}(\boldsymbol{x}), \ldots\}$进行广义离散傅立叶变换，实现这种变换要受到函数 $\mathbf{A}^{(k)}(\boldsymbol{x})$中的项的数量的制约，即当 $\mathbf{A}^{(k)}(\boldsymbol{x})$ 关于系数的函数规模发生爆炸时，将大大增加进行广义离散傅立叶变换的代价。

综上所述，本发明对于密码的安全性来说，实现了一种质的飞跃。

发明人作出本发明之后，在与背景技术中提及的三种现有技术相比较中发现：在本发明降低安全性的极端情况下，其数学表达与 DH 相近似。即，当本发明基于 ***n*=1 的 1 变元单项式函数**的迭代层数问题时，其迭代函数为：$f(x)=gx$，二层迭代的结果为 $f(f(x))=g^2x$，…，k 层迭代的结果为 $f(\ldots(f(x))\ldots)=g^kx$；**再令 *x*=1**，则从表面现象看，此时本发明的数学表述与 DH 相近似。为了避免他人在看到本发明之后，将本发明简单的理解为 DH 的自然拓展，**下面对本发明的创新难点进行简单说明。**

1、从DH方案看，对本发明没有任何的提示。

首先，现有技术已经认识到背景技术中提及的三种编码体制存在不安全的可能性，但是其具体导致不安全的因素是什么？改进的具体方向是什么？具体如何改进？以及应该以哪个编码体制为基础进行改进？这些问题，现有技术没有给出任何的提示。

其次，当将本发明限定在***n*=1**、**单项式函数以及*x*=1**时，本发明和DH在数学表述上确实相近似，但是本领域技术人员都应该知悉，实际上，$d = g^k \bmod p$ 可以是非常多的数学模型的极端情况，而从一个极端情况推导出其确定的普遍形式，是几乎不可能的事情。例如，也可以把基本域$\mathbf{F}_p$的离散对数问题扩展为基本域$\mathbf{F}_p$上的方阵的离散对数问题等。也可以说，本发明是从$d = g^k \bmod p$所对应的非常多的数学模型中选择出来一个最合适的数学模型。

最重要的是，实际上是本发明第一次提出了迭代函数保形的概念，从DH来看，其中没有变元***x*，**也没有该变元***x***的函数，进而难以想到函数的迭代，更不要说提示发明人需要考虑迭代函数是否能够保形，以及保形有什么意义。

事实上，本发明提出了一种完全崭新的研究方向，只不过恰好在数学表述

上的极端形式上与DH比较相似。在现有的各种公开文献中，不仅“保形迭代”的术语是本发明首次使用的，其概念、定义、性质描述、判定方法、建立步骤等，也是本发明首次提出的。

2、作出本发明，需要克服长久以来的技术偏见。

建立关于 x 的函数规模不扩张、而关于其系数的函数规模却发生爆炸的多变元非线性迭代变换，一直被认为是一个很难实现的问题。首先，把简单的单变元单项式的迭代，扩展为单变元多项式的非线性迭代，必然会遇到函数扩张问题。例如：设迭代函数为：$f(x) = (a_0 + a_1x + a_2x^2) \bmod p$。当 k=2 时：

$f(f(x)) = (a_2(a_0 + a_1x + a_2x^2)^2 + a_1(a_0 + a_1x + a_2x^2) + a_0) \bmod p$

$= (a_0 + a_0a_1 + a_0^2a_2 + a_1^2x + 2a_0a_1a_2x + a_1a_2x^2 + a_1^2a_2x^2 + 2a_0a_2^2x^2 + 2a_1a_2^2x^3 + a_2^3x^4) \bmod p$，此时出现了 x^3、x^4 等在 $f(x)$中不存在的项，经过多层迭代后上述函数的规模必将发生爆炸。用这种函数完成加密运算，需要指数级的巨大存储空间和漫长的计算时间。

其次，把单变元函数的迭代，简单地扩展为多变元函数的迭代，通常也会遇到函数扩张问题。例如：$(x_1^2 + x_2，x_1 + x_2)$，经过二层迭代后为：$(x_1^4 + 2x_1^2x_2 + x_2^2 + x_1 + x_2，x_1^2 + x_1 + 2x_2)$，新出现了关于 x_1^4、$x_1^2x_2$、x_2^2 等原先不存在的项。

正是由于上述的情况，长久以来，本领域技术人员公认：对于具有 2 个以上个项的一元非线性函数，经过迭代后必然导致函数的规模发生组合爆炸。因此，对于多元非线性函数，人们也普遍认为，经过迭代后也应该导致函数的规模发生组合爆炸。

然而并没有理论证明：“对于具有 2 个以上个项的一元非线性函数，经过迭代后必然导致函数的规模发生组合爆炸”，**这个结论一定可以推广到二元以上的情况：也就是说，对于多元非线性函数，经过迭代后也必然导致函数的规模发生组合爆炸。**因此，本发明的提出首先要克服几十年来形成的这种技术偏见。发明人需要确定：在多元非线性函数中，存在既能够保证关于变元$(x_1, \ldots, x_n)$的非线性次数保持不变、又能够保证关于该函数中的系数的非线性次数迅速增加的函数，并且，这种函数可以用确定的方法建立起来。

3、实现本发明，建立多变元非线性保形迭代变换有很高的难度。

首先，本发明属于开拓性研究，其数学理论背景不成熟，可供借鉴的文

献资料非常少，例如：怎样从抽象空间的角度来理解保形迭代变换的数学结构？怎样建立从有理分式扩域到多项式扩域的同态映射、以及从多项式扩域到基本域的同态映射？怎样求保形迭代的周期？如何确定保形迭代具体的数学性质，以及如何判定这些性质？这些问题涉及到一些深刻的、目前尚未完全解决的数学前沿课题。

其次，提出本发明的概念很容易，但要设计出可行的、实用的、完整的技术解决方案，却需要很高的技术门槛：不仅要把握当代数学前沿的进展，还要有丰富的实际编码经验和分析水平，能熟练地运用数学工具，此外还要依赖于灵感、机遇等非确定因素，本领域一般技术人员很难完成这项工作，例如：

对于第一类保形迭代变换，通过用两个线性多项式相除来建立有理分式，看起来很简单，但要理解其算法背后的原理，涉及到复杂的数学推导；

对于第二类、第三类保形迭代变换，还涉及到解不定方程组的技术手段、保形迭代性质的判定方法问题，需要进行复杂的符号运算和定量分析。

现有的 DH 数字签名算法都需要计算密码周期，但保形迭代变换的周期（即 $\mathbf{A}^{(k)}(\boldsymbol{x})=\mathbf{A}(\boldsymbol{x})$时的 k）通常是难以计算的，如何在数字签名中绕过复杂的周期计算，需要相当高的编码技巧。本发明主要是通过整数的计算，而不是模一个周期的计算，来建立签名方程。需要指出的是，虽然这个周期问题在密钥协商、加密和签名中都存在，但在签名时问题尤其突出。

4、对于“保形迭代”这种函数性质能够带来怎样的有益效果需要对密码的规律和本质有深刻的认识，才能获得充分地理解。

例如，对于保形迭代所产生的“分形”（fractal）效果，就需要发挥想象力才能感悟到其算法设计的巧妙之处。具体的，在上述的n=2的第一类$\mathbf{A}(\boldsymbol{x})$中，$\mathbf{A}^{(k+1)}(\boldsymbol{x})$相对于$\mathbf{A}^{(k)}(\boldsymbol{x})$来说，从$\mathbf{A}^{(k)}(\boldsymbol{x})$的未知元$x_1$、$x_2$的局部看进去，在$\mathbf{A}^{(k+1)}(\boldsymbol{x})$中的对应位置上，都存在着一个与$\mathbf{A}(\boldsymbol{x})$相似的函数结构。这种描述不仅具有纯理论的、美妙的艺术观赏价值，而且是一种实质性的密码设计：当迭代层数增加时，尚可以想象其函数规模的爆炸方式具有某种规律性，然而只要把$\mathbf{A}^{(k)}(\boldsymbol{x})$中的系数代入具体的值并展开、化简，这种规律性，即函数的结构信息，就会消失。

总之，如何建立满足密码学性质的既具有很强的非线性、又不会带来关于

$\boldsymbol{x}$ 的函数规模爆炸的保形迭代函数，是一项探索性极强的前沿课题，它的研究既有广泛的应用前景，又有很高的技术难度，经历了从理论到实践的反复过程，是发明人长期思考的结果，充分体现了发明人的智慧创新。

本发明与现有技术中的 ECC 公钥密码体制相比，主要区别在于：

首先是数学概念不同：椭圆曲线上的点用一个二维数组(x, y)来表示，椭圆曲线群定义了一种“加法”——这是一种由一条椭圆曲线中的两个点、求第三个点的非线性运算，但这种运算不满足保形迭代函数的定义。

其次是所基于的数学难题不同：一个保形迭代变换 $\mathbf{A}(\boldsymbol{x})$等价于一个 n 输入、n 输出的函数，集合$\{\mathbf{A}^{(1)}(\boldsymbol{x}), \mathbf{A}^{(2)}(\boldsymbol{x}), \ldots, \mathbf{A}^{(k)}(\boldsymbol{x}), \ldots\}$对于迭代运算来说组成一个半群。所谓保形迭代层数问题，可理解为在该半群中定义了一种“保形迭代离散对数问题”，其数学性质与“椭圆曲线离散对数问题”有很大区别。

最后是密码周期不同：目前尚未发现计算保形迭代的周期（该周期定义为$\mathbf{A}^{(k)}(\boldsymbol{x})=\mathbf{A}(\boldsymbol{x})$时的 k）的通用方法，为此本发明有意地避开了困难的周期计算问题；而 ECC 的周期（即椭圆曲线上的点的阶）是可以计算的。

由于本发明与 ECC 的算法原理不同，缺少可比性，所以从算法空间的角度来理解本发明的有益效果：ECC 采用两个点之间的值的运算，其算法空间对应于一个二维平面上的椭圆曲线的点的集合，该集合中的元素是用二维向量(x, y)的值来表示的；而本发明采用两个函数之间的算子的运算，其算法空间对应于一个多项式组或有理分式组的集合，从抽象空间的角度：该集合中的元素是用该函数组中的系数来表示的，与它们的未知元 $x_1, \ldots, x_n$ 的值无关；例如，多项式组$((a_0 + a_1x_1 + a_2x_1^2 + a_3x_1x_2) \bmod p, (b_0 + b_1x_1 + b_2x_1^2 + b_3x_1x_2) \bmod p)$是用系数 $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$ 来描述的，与(x_1, x_2)的值无关，属于 $\mathbf{F}_p$ 上的 8 维空间的点的集合；显然，本发明具有更大的算法空间，其变化规律也更加复杂。

以上对本发明所提供的一种密钥协商的方法、一种用于编码和译码数字消息的方法和系统，以及一种用于数字签名的方法和系统，进行了详细介绍，本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本发明的限制。

权 利 要 求

1、一种密钥协商的方法，其特征在于，包括：

步骤 1，预置用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x}))=\mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2，所述用户群中的各用户互相交换各自与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果；

步骤 3，各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

2、如权利要求 1 所述的方法，其特征在于，当所述用户群仅包括两个用户时，所述步骤 2 进一步包括：

第一用户选择整数 k_1，计算第一中间结果，并传递至第二用户；所述第一中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_1 层迭代有关；

第二用户选择整数 k_2，计算第二中间结果，并传递至第一用户；所述第二中间结果与 $\mathbf{A}(\boldsymbol{x})$的 k_2 层迭代有关。

3、如权利要求 1 所述的方法，其特征在于，还包括：建立所述用户群共享的、变元数量大于 1 的向量 $\boldsymbol{q}$；并且，所述用户群仅包括两个用户，则

所述步骤 2 进一步包括：第一用户选择整数 k_1，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_1 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_1=\mathbf{A}^{(k_1)}(\boldsymbol{q})$，将计算结果 $\boldsymbol{d}_1$ 传递给第二用户；第二用户选择整数 k_2，把 $\boldsymbol{q}$ 代入 $\mathbf{A}(\boldsymbol{x})$并进行 k_2 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\boldsymbol{d}_2=\mathbf{A}^{(k_2)}(\boldsymbol{q})$，将计算结果 $\boldsymbol{d}_2$ 传递给第一用户；

所述步骤 3 进一步包括：第一用户计算密钥 $\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(k_1)}(\boldsymbol{d}_2)$；第二用户计算密钥 $\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(k_2)}(\boldsymbol{d}_1)$；

其中，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k_1)}(\mathbf{A}^{(k_2)}(\boldsymbol{x}))=\mathbf{A}^{(k_1+k_2)}(\boldsymbol{x})$。

4、如权利要求 1 所述的方法，其特征在于，当所述用户群仅包括两个用户时，

所述步骤 2 进一步包括：第一用户选择整数 k_1，计算 k_1 层 $\mathbf{A}(\boldsymbol{x})$的迭代：

$\mathbf{B}_1(\boldsymbol{x})=\mathbf{A}^{(k_1)}(\boldsymbol{x})$，并把函数组 $\mathbf{B}_1(\boldsymbol{x})$传递给用户 2；第二用户选择整数 k_2，计算 k_2 层 $\mathbf{A}(\boldsymbol{x})$的迭代：$\mathbf{B}_2(\boldsymbol{x})=\mathbf{A}^{(k_2)}(\boldsymbol{x})$，并把函数组 $\mathbf{B}_2(\boldsymbol{x})$传递给第一用户；

所述步骤 3 进一步包括：第一用户计算密钥 $\boldsymbol{K}=\mathbf{B}_2{}^{(k_1)}(\boldsymbol{x})$；第二用户计算密钥 $\boldsymbol{K}=\mathbf{B}_1{}^{(k_2)}(\boldsymbol{x})$；

其中，所述 A(x)进一步满足：若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k_1)}(\boldsymbol{x})$，则 $\mathbf{B}^{(k_2)}(\boldsymbol{x})=\mathbf{A}^{(k_1k_2)}(\boldsymbol{x})$。

5、如权利要求 1 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个域 $\mathbf{F}$ 上的 n 元有理分式函数组成，其每个有理分式函数中的分子、分母均为关于$(x_1, \ldots, x_n)$的线性多项式，其分母多项式相同；

接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n 和变元的数据长度；

生成 $\mathbf{A}(\boldsymbol{x})$中的每个项的系数；

按照预置结构，输出得到的 $\mathbf{A}(\boldsymbol{x})$。

6、如权利要求 1 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

a、预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个域 $\mathbf{F}$ 上的 n 元有理函数组成，其含有关于$(x_1, \ldots, x_n)$的大于 1 次的项；当 $A_i(x_1, \ldots, x_n)$的分母为 0 次多项式时，所述有理函数为多项式；当 $A_i(x_1, \ldots, x_n)$的分母为大于 1 次的多项式时，所述有理函数为有理分式；

b、接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 $\mathbf{A}(\boldsymbol{x})$的表示形式，所述 $\mathbf{A}(\boldsymbol{x})$中的不为零的系数用变元符号表示；

d、将 $\mathbf{A}(\boldsymbol{x})$代入自身并执行展开、化简的数据处理：$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$；

e、针对 $\mathbf{B}(\boldsymbol{x})$与 $\mathbf{A}(\boldsymbol{x})$对比而新出现的每一个关于 $\boldsymbol{x}$ 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断所述方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到该方程组的一组解，并作为 $\mathbf{A}(\boldsymbol{x})$中的系数的值，代入步骤 c 生成的 $\mathbf{A}(\boldsymbol{x})$的表示形式；

g、输出所得到的 $\mathbf{A}(\boldsymbol{x})$。

7、如权利要求 1 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

a、预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个环 $\mathbf{R}$ 上的 n 元多项式组成：其含有关于 $(x_1, \ldots, x_n)$的大于 1 次的项；

b、接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 $\mathbf{A}(\boldsymbol{x})$的表示形式，所述 $\mathbf{A}(\boldsymbol{x})$中的不为零的系数用变元符号表示；

d、将 $\mathbf{A}(\boldsymbol{x})$代入自身并执行展开、化简的数据处理：$\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$；

e、针对 $\mathbf{B}(\boldsymbol{x})$与 $\mathbf{A}(\boldsymbol{x})$对比而新出现的每一个关于 $\boldsymbol{x}$ 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到该方程组的一组解，并作为 $\mathbf{A}(\boldsymbol{x})$中的系数的值，代入步骤 c 生成的 $\mathbf{A}(\boldsymbol{x})$的表示形式；

g、输出所得到的 $\mathbf{A}(\boldsymbol{x})$。

8、如权利要求 6 或 7 所述的方法，其特征在于，在所述步骤 d 和步骤 e 之间还包括：

将 $\mathbf{B}(\boldsymbol{x})$与 $\mathbf{A}(\boldsymbol{x})$对比，若 $\mathbf{B}(\boldsymbol{x})$中的关于 $\boldsymbol{x}$ 的每一种新出现的项，都至少有两个，则执行步骤 e，否则返回到步骤 c。

9、如权利要求 1 所述的方法，其特征在于，还包括：

依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；

采用该伪随机序列的种子，标识该 $\mathbf{A}(\boldsymbol{x})$。

10、一种用于编码和译码数字消息的方法，其特征在于，包括：

步骤 1，预置加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2、选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

步骤 3、加密端选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，

然后利用该中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

步骤 4、解密端利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，然后利用所述中间密钥对加密结果进行解密。

11、如权利要求 10 所述的方法，其特征在于，还包括：建立加密端和解密端共享的、变元数量大于 1 的向量 $\boldsymbol{q}$，公钥 $\boldsymbol{d}=(d_1, \ldots, d_n)=\mathbf{A}^{(k)}(\boldsymbol{q})$；则

所述步骤 3 进一步包括：加密端选择整数 t，将公钥变换为关于 t 的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(t)}(\boldsymbol{d})$，然后利用所述中间密钥 $\boldsymbol{K}$ 对明文 M 进行加密，$C=\mathrm{D}(M, \boldsymbol{K})$，传送包含加密结果 C 和 t 的变换结果 $\boldsymbol{v}$ 的密文 E 至解密端，$E=\{\boldsymbol{v}, C\}$，$\boldsymbol{v}=(v_1, \ldots, v_n)=\mathbf{A}^{(t)}(\boldsymbol{q})$；

所述步骤 4 进一步包括：解密端利用 t 的变换结果 $\boldsymbol{v}$、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=(K_1, \ldots, K_n)=\mathbf{A}^{(k)}(\boldsymbol{v})$，然后利用所述中间密钥 $\boldsymbol{K}$ 对加密结果 C 进行解密，得到明文 M，$M=\mathrm{D}^{-1}(C, \boldsymbol{K})$；

其中，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k)}(\mathbf{A}^{(t)}(\boldsymbol{x}))=\mathbf{A}^{(k+t)}(\boldsymbol{x})$。

12、如权利要求 10 所述的方法，其特征在于，当公钥 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}^{(k)}(\boldsymbol{x})$时，则

所述步骤 3 进一步包括：加密端选择整数 t，将公钥变换为关于 t 的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=\mathbf{B}^{(t)}(\boldsymbol{x})$，然后利用所述中间密钥 $\boldsymbol{K}$ 对明文 M 进行加密，$C=\mathrm{D}(M, \boldsymbol{K})$，传送包含加密结果 C 和 t 的变换结果 $\mathbf{V}(\boldsymbol{x})$的密文 E 至解密端，$E=\{\mathbf{V}(\boldsymbol{x}), C\}$，$\mathbf{V}(\boldsymbol{x})=\mathbf{A}^{(t)}(\boldsymbol{x})$；

所述步骤 4 进一步包括：解密端利用 t 的变换结果 $\mathbf{V}(\boldsymbol{x})$、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥 $\boldsymbol{K}$，$\boldsymbol{K}=\mathbf{V}^{(k)}(\boldsymbol{x})$，然后利用所述中间密钥 $\boldsymbol{K}$ 对加密结果 C 进行解密，得到明文 M，$M=\mathrm{D}^{-1}(C, \boldsymbol{K})$；

其中，所述 A(x)进一步满足：若 B(x) = A(k)(x)，则 B(t)(x) = A(k t)(x)。

13、如权利要求 10 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个域 $\mathbf{F}$ 上的 n 元有理分式函数组成，其每个有理分式函数中的分子、分母均为关于$(x_1, \ldots, x_n)$的线性多项式，其分母多项式相同；

接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n 和变元的数据长度；

生成 **A**(***x***)中的每个项的系数；

按照预置结构，输出得到的 **A**(***x***)。

14、如权利要求 10 所述的方法，其特征在于，通过以下步骤建立 *n* 元非线性函数组 **A**(***x***)：

a、预置 **A**(***x***)的结构：**A**(***x***)由 *n* 个域 **F** 上的 *n* 元有理函数组成，其含有关于(x_1, …, x_n)的大于 1 次的项；当 A_i(x_1, …, x_n)的分母为 0 次多项式时，所述有理函数为多项式；当 A_i(x_1, …, x_n)的分母为大于 1 次的多项式时，所述有理函数为有理分式；

b、接收 **A**(***x***)的相关技术指标参数，所述指标参数包括变元数量 *n*、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 **A**(***x***)的表示形式，所述 **A**(***x***)中的不为零的系数用变元符号表示；

d、将 **A**(***x***)代入自身并执行展开、化简的数据处理：**B**(***x***) = **A**(**A**(***x***))；

e、针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到该方程组的一组解，并作为 **A**(***x***)中的系数的值，代入步骤 c 生成的 **A**(***x***)的表示形式；

g、输出所得到的 **A**(***x***)。

15、如权利要求 10 所述的方法，其特征在于，通过以下步骤建立 *n* 元非线性函数组 **A**(***x***)：

a、预置 **A**(***x***)的结构：**A**(***x***)由 *n* 个环 **R** 上的 *n* 元多项式组成：其含有关于(x_1, …, x_n)的大于 1 次的项；

b、接收 **A**(***x***)的相关技术指标参数，所述指标参数包括变元数量 *n*、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 **A**(***x***)的表示形式，所述 **A**(***x***)中的不为零的系数用变元符号表示；

d、将 **A**(***x***)代入自身并执行展开、化简的数据处理：**B**(***x***) = **A**(**A**(***x***))；

e、针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计

算得到该方程组的一组解，并作为 **A**(***x***)中的系数的值，代入步骤 c 生成的 **A**(***x***)的表示形式；

g、输出所得到的 **A**(***x***)。

16、如权利要求 14 或 15 所述的方法，其特征在于，在所述步骤 d 和步骤 e 之间还包括：

将 **B**(***x***)与 **A**(***x***)对比，若 **B**(***x***)中的关于 ***x*** 的每一种新出现的项，都至少有两个，则执行步骤 e，否则返回到步骤 c。

17、如权利要求 10 所述的方法，其特征在于，还包括：

依据伪随机序列确定 **A**(***x***)中的系数的值；

采用所述伪随机序列的种子，标识所述 **A**(***x***)。

18、如权利要求 10 所述的方法，其特征在于，通过以下步骤建立私钥：

预置λ个私钥表$L_1, \ldots, L_\lambda$以及对应的公钥表$\mathbf{G}_1, \ldots, \mathbf{G}_\lambda$，分布在$\lambda$个密钥分配中心；

依据预置规则，根据用户的身份ID获得指向多个私钥表的指针；

分别从所指向的多个私钥表中各获取一个或者多个私钥分量，组合得到该用户的私钥。

19、一种用于数字签名及验证的方法，其特征在于，包括：

步骤 1，预置签名端和验证端共享的 **A**(***x***)；所述 **A**(***x***)是由 *n* 元向量 ***x*** 到 *n* 元向量 ***y*** 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n)=\mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 **A**(***x***)需要满足：把 **A**(***x***)的 *s* 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 **A**(***x***)相比，其关于 ***x*** 的系数不为 0 的项的数量与类型保持不变，*s* 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x}))=\mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

步骤 2、选择整数 *k* 作为私钥；运用 **A**(***x***)的 *k* 层迭代建立对应的公钥；

步骤 3、签名端选择整数 *t*，依据预置规则将待签名数据变换为与 *t*、私钥 *k* 相关的中间消息，然后传送包含中间消息和 *t* 的变换结果的数字签名至验证端；所述 *t* 的变换结果与 **A**(***x***)的 *t* 层迭代相关；

步骤 4、验证端利用 *t* 的变换结果、待签名数据、中间消息、公钥和 **A**(***x***)验证是否满足所述预置规则，如果满足，则所述数字签名验证通过。

20、如权利要求 19 所述的方法，其特征在于，还包括：

直接验证是否满足所述预置规则；

或者，对该预置规则进行变换，通过验证是否满足变换后的预制规则来验证签名是否正确。

21、如权利要求 19 所述的方法，其特征在于，还包括：建立签名端和验证端共享的、变元数量大于 1 的向量 $\boldsymbol{q}$；则

所述步骤 3 进一步包括：签名端选择整数 t，依据预置规则将待签名数据 M 变换为与 t、私钥 k 相关的中间消息 c，然后传送包含中间消息 c 和 t 的变换结果 $\boldsymbol{e}$ 的数字签名 S 至解密端，$S = \{c, \boldsymbol{e}\}$；所述 t 的变换结果 $\boldsymbol{e}$ 与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关：$\boldsymbol{e} = (e_1, \ldots, e_n) = \mathbf{A}^{(t)}(\boldsymbol{q})$；其中，所述预置规则为整数方程：$c = \Phi(t, w, k)$，$w$ 为依据待签名数据 M 计算得到的整数；

所述步骤 4 进一步包括：验证端利用 t 的变换结果 $\boldsymbol{e}$、依据待签名数据 M 计算得到的 w、中间消息 c、公钥和 $\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则：假设整数方程 Φ 可进一步表示为: $\alpha = \beta$, 并且β中包含 t, 则验证 $\mathbf{A}^{(\alpha)}(\boldsymbol{q}) = \mathbf{A}^{(\beta)}(\boldsymbol{q}) = \mathbf{A}^{(\beta - t)}(\boldsymbol{e})$ 是否成立；如果成立，则所述数字签名验证通过；

其中，当公钥 $\boldsymbol{d} = (d_1, \ldots, d_n) = \mathbf{A}^{(k)}(\boldsymbol{q})$时，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：$\mathbf{A}^{(k)}(\mathbf{A}^{(t)}(\boldsymbol{x})) = \mathbf{A}^{(k+t)}(\boldsymbol{x})$；当公钥 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}^{(k)}(\boldsymbol{x})$时，所述 $\mathbf{A}(\boldsymbol{x})$进一步满足：若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}^{(k)}(\boldsymbol{x})$，则 $\mathbf{B}^{(t)}(\boldsymbol{x}) = \mathbf{A}^{(k\,t)}(\boldsymbol{x})$。

22、如权利要求 19 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个域 $\mathbf{F}$ 上的 n 元有理分式函数组成，其每个有理分式函数中的分子、分母均为关于$(x_1, \ldots, x_n)$的线性多项式，其分母多项式相同；

接收 $\mathbf{A}(\boldsymbol{x})$的相关技术指标参数，所述指标参数包括变元数量 n 和变元的数据长度；

生成 $\mathbf{A}(\boldsymbol{x})$中的每个项的系数；

按照预置结构，输出得到的 $\mathbf{A}(\boldsymbol{x})$ 。

23、如权利要求 19 所述的方法，其特征在于，通过以下步骤建立 n 元非线性函数组 $\mathbf{A}(\boldsymbol{x})$：

a、预置 $\mathbf{A}(\boldsymbol{x})$的结构：$\mathbf{A}(\boldsymbol{x})$由 n 个域 $\mathbf{F}$ 上的 n 元有理函数组成，其含有关于$(x_1, \ldots, x_n)$的大于 1 次的项；当 $A_i(x_1, \ldots, x_n)$的分母为 0 次多项式时，所述有理函数为多项式；当 $A_i(x_1, \ldots, x_n)$的分母为大于 1 次的多项式时，所述有理函数为有理分式；

b、接收 **A**(***x***)的相关技术指标参数，所述指标参数包括变元数量 *n*、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 **A**(***x***)的表示形式，所述 **A**(***x***)中的不为零的系数用变元符号表示；

d、将 **A**(***x***)代入自身并执行展开、化简的数据处理：**B**(***x***) = **A**(**A**(***x***))；

e、针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到该方程组的一组解，并作为 **A**(***x***)中的系数的值，代入步骤 c 生成的 **A**(***x***)的表示形式；

g、输出所得到的 **A**(***x***)。

24、如权利要求 19 所述的方法，其特征在于，通过以下步骤建立 *n* 元非线性函数组 **A**(***x***)：

a、预置 **A**(***x***)的结构：**A**(***x***)由 *n* 个环 **R** 上的 *n* 元多项式组成：其含有关于 $(x_1, \ldots, x_n)$的大于 1 次的项；

b、接收 **A**(***x***)的相关技术指标参数，所述指标参数包括变元数量 *n*、变元的数据长度和最高的非线性次数；

c、依据所述指标参数和预置结构，生成一个 **A**(***x***)的表示形式，所述 **A**(***x***)中的不为零的系数用变元符号表示；

d、将 **A**(***x***)代入自身并执行展开、化简的数据处理：**B**(***x***) = **A**(**A**(***x***))；

e、针对 **B**(***x***)与 **A**(***x***)对比而新出现的每一个关于 ***x*** 的项，生成关于这些项的系数的多项式，令这些多项式的值为 0，从而建立联立方程组；

f、判断该方程组是否有解，如果没有解，则返回步骤 c；如果有解，则计算得到所述方程组的一组解，并作为 **A**(***x***)中的系数的值，代入步骤 c 生成的 **A**(***x***)的表示形式；

g、输出所得到的 **A**(***x***)。

25、如权利要求 23 或 24 所述的方法，其特征在于，在所述步骤 d 和步骤 e 之间还包括：

将 **B**(***x***)与 **A**(***x***)对比，若 **B**(***x***)中的关于 ***x*** 的每一种新出现的项，都至少有两个，则执行步骤 e，否则返回到步骤 c。

26、如权利要求 19 所述的方法，其特征在于，还包括：

依据伪随机序列确定 $\mathbf{A}(\boldsymbol{x})$中的系数的值；

采用所述伪随机序列的种子，标识所述 $\mathbf{A}(\boldsymbol{x})$。

27、如权利要求 19 所述的方法，其特征在于，通过以下步骤建立私钥：

预置λ个私钥表$L_1, \ldots, L_\lambda$以及对应的公钥表$\mathbf{G}_1, \ldots, \mathbf{G}_\lambda$，分布在$\lambda$个密钥分配中心；

依据预置规则，根据用户的身份ID获得指向多个私钥表的指针；

分别从所指向的多个私钥表中各获取一个或者多个私钥分量，组合得到所述用户的私钥。

28、一种密钥协商的系统，其特征在于，包括：

共享单元，用于存储用户群共享的 $\mathbf{A}(\boldsymbol{x})$，所述用户群包括至少两个用户；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x}) = \mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

中间结果交换单元，连接所述用户群中的各用户端，用于传递所述用户群中的各用户与 $\mathbf{A}(\boldsymbol{x})$的整数层迭代有关的中间结果至其他用户；

密钥计算单元，位于所述用户群中的各用户端，用于针对各用户分别利用所接收的中间结果，计算得到该用户群共享的密钥 $\boldsymbol{K}$。

29、一种用于编码和译码数字消息的系统，其特征在于，包括：

共享单元，用于存储加密端和解密端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

加密单元，位于加密端，用于选择整数 t，运用 $\mathbf{A}(\boldsymbol{x})$将公钥变换为关于 t 的中间密钥，利用所述中间密钥对明文进行加密，传送加密结果和 t 的变换结果至解密端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

解密单元，位于解密端，用于利用 t 的变换结果、私钥 k 和 $\mathbf{A}(\boldsymbol{x})$计算得到相同的中间密钥，利用所述中间密钥对加密结果进行解密。

30、一种用于数字签名及验证的系统，其特征在于，包括：

共享单元，用于存储签名端和验证端共享的 $\mathbf{A}(\boldsymbol{x})$；所述 $\mathbf{A}(\boldsymbol{x})$是由 n 元向量 $\boldsymbol{x}$ 到 n 元向量 $\boldsymbol{y}$ 的非线性函数组

$$\boldsymbol{y}=(y_1, \ldots, y_n) = \mathbf{A}(\boldsymbol{x})=(\mathrm{A}_1(x_1, \ldots, x_n), \ldots, \mathrm{A}_n(x_1, \ldots, x_n))$$

其中，$n>1$，所述 $\mathbf{A}(\boldsymbol{x})$需要满足：把 $\mathbf{A}(\boldsymbol{x})$的 s 层迭代 $\mathbf{A}^{(s)}(\boldsymbol{x})$，与 $\mathbf{A}(\boldsymbol{x})$相比，其关于 $\boldsymbol{x}$ 的系数不为 0 的项的数量与类型保持不变，s 为整数；若 $\mathbf{B}(\boldsymbol{x})=\mathbf{A}(\mathbf{A}(\boldsymbol{x}))$，则 $\mathbf{A}(\mathbf{B}(\boldsymbol{x})) = \mathbf{B}(\mathbf{A}(\boldsymbol{x}))$；

公私钥建立单元，用于选择整数 k 作为私钥；运用 $\mathbf{A}(\boldsymbol{x})$的 k 层迭代建立对应的公钥；

签名单元，位于签名端，用于选择整数 t，依据预置规则将待签名数据变换为与 t、私钥 k 相关的中间消息，传送包含中间消息和 t 的变换结果的数字签名至验证端；所述 t 的变换结果与 $\mathbf{A}(\boldsymbol{x})$的 t 层迭代相关；

验证单元，位于验证端，用于利用 t 的变换结果、待签名数据、中间消息、公钥和 $\mathbf{A}(\boldsymbol{x})$验证是否满足所述预置规则，如果满足，则该数字签名验证通过。

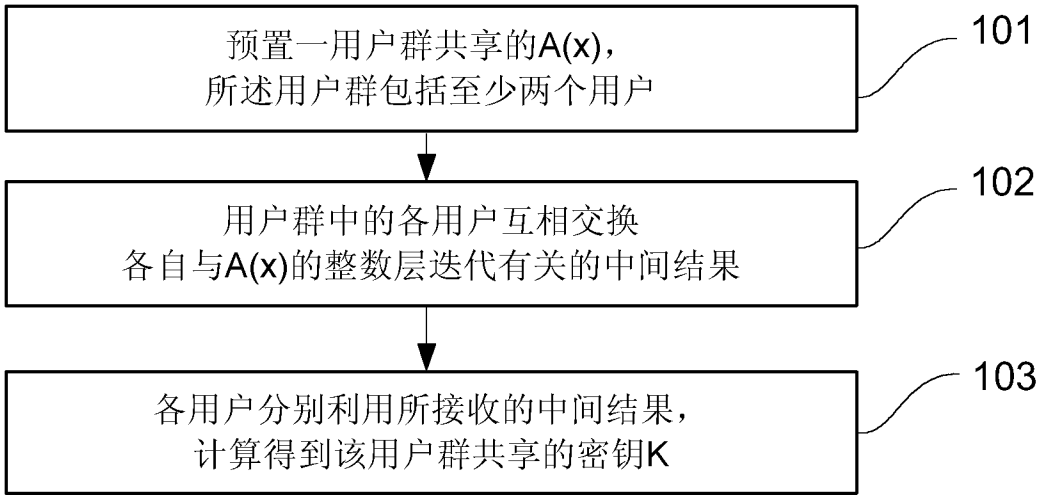
预置一用户群共享的A(x)，
所述用户群包括至少两个用户
101
用户群中的各用户互相交换
各自与A(x)的整数层迭代有关的中间结果
102
各用户分别利用所接收的中间结果，
计算得到该用户群共享的密钥K
103

图 1

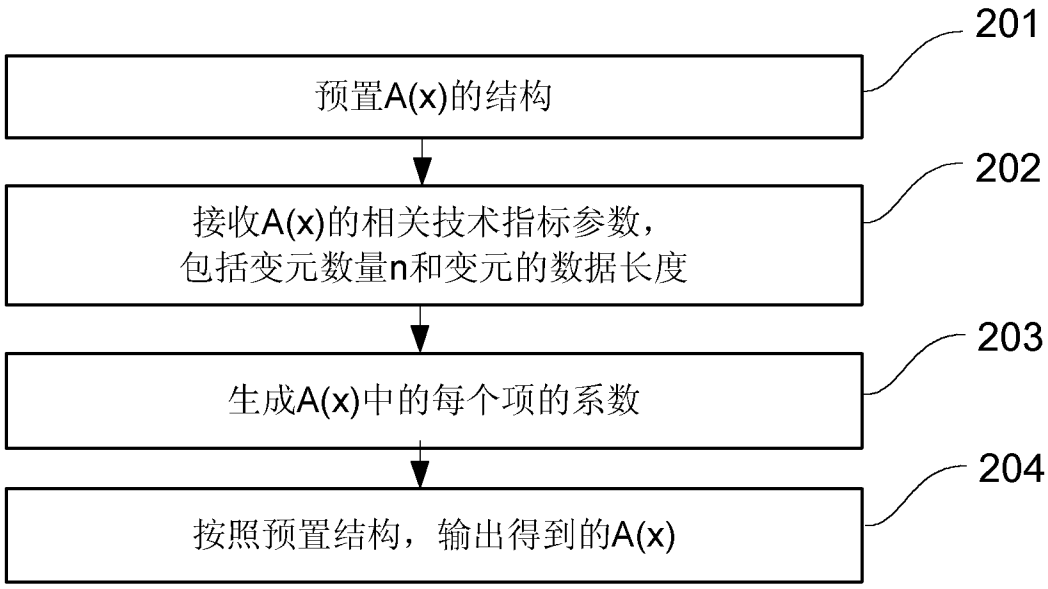
预置A(x)的结构
201
接收A(x)的相关技术指标参数，
包括变元数量n和变元的数据长度
202
生成A(x)中的每个项的系数
203
按照预置结构，输出得到的A(x)
204

图 2


301
预置A(x)的结构
302
接收A(x)的相关技术指标参数，所述指标参数包括变元数量n、变元的数据长度和最高的非线性次数
303
依据所述指标参数和预置结构，生成一个A(x)的表示形式，该A(x)中的不为零的系数用变元符号表示
304
将A(x)代入自身并执行展开、化简的数据处理：B(x) = A(A(x))
305
针对B(x)与A(x)对比而新出现的每一个关于x的项，生成关于这些项的系数的多项式，令这些多项式的值为0，从而建立联立方程组
306
判断该方程组是否有解，如果没有解，则返回步骤303；如果有解，则计算得到该方程组的一组解，并作为A(x)中的系数的值，代入步骤303生成的A(x)的表示形式
307
输出所得到的A(x)

图 3

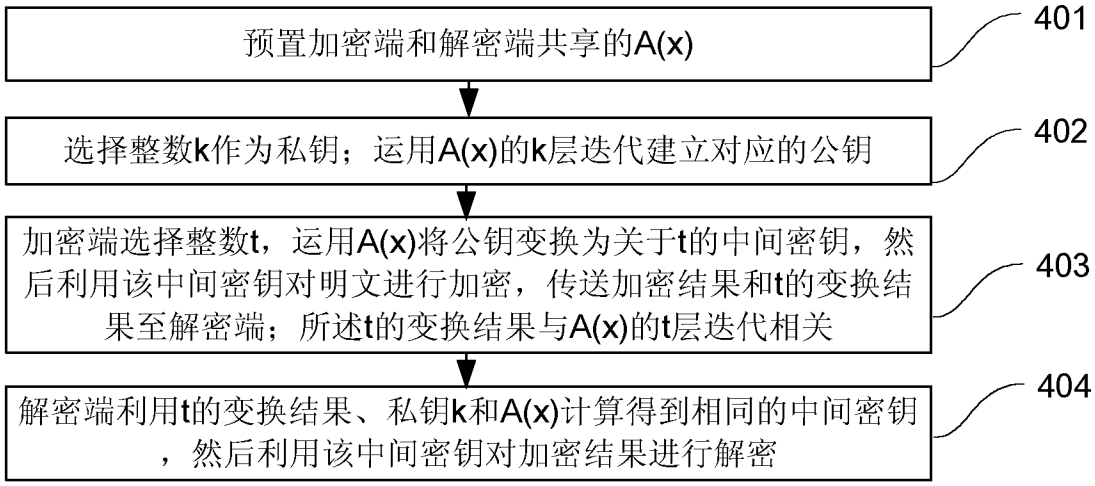
预置加密端和解密端共享的A(x)
401
选择整数k作为私钥；运用A(x)的k层迭代建立对应的公钥
402
加密端选择整数t，运用A(x)将公钥变换为关于t的中间密钥，然后利用该中间密钥对明文进行加密，传送加密结果和t的变换结果至解密端；所述t的变换结果与A(x)的t层迭代相关
403
解密端利用t的变换结果、私钥k和A(x)计算得到相同的中间密钥，然后利用该中间密钥对加密结果进行解密
404

图 4

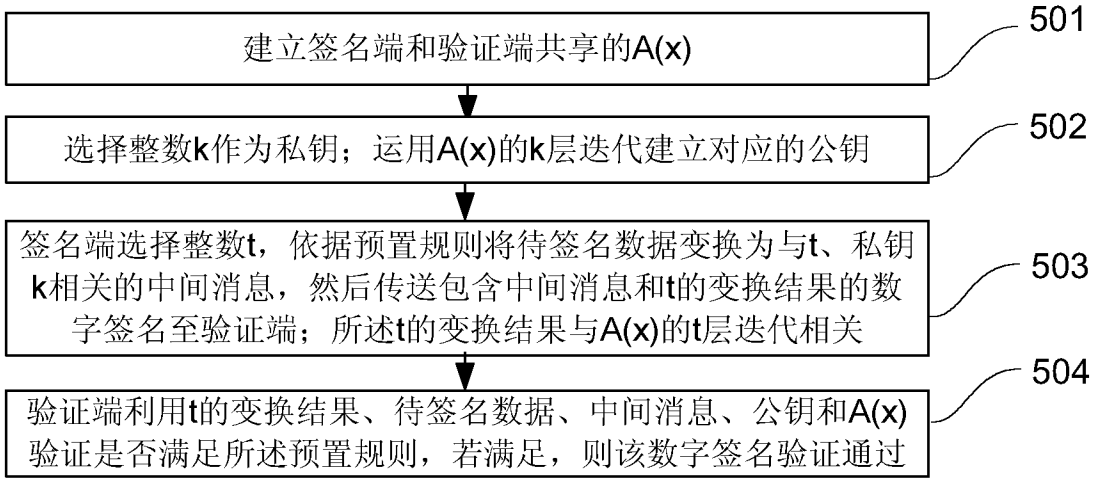
建立签名端和验证端共享的A(x)
501
选择整数k作为私钥；运用A(x)的k层迭代建立对应的公钥
502
签名端选择整数t，依据预置规则将待签名数据变换为与t、私钥k相关的中间消息，然后传送包含中间消息和t的变换结果的数字签名至验证端；所述t的变换结果与A(x)的t层迭代相关
503
验证端利用t的变换结果、待签名数据、中间消息、公钥和A(x)验证是否满足所述预置规则，若满足，则该数字签名验证通过
504

图 5

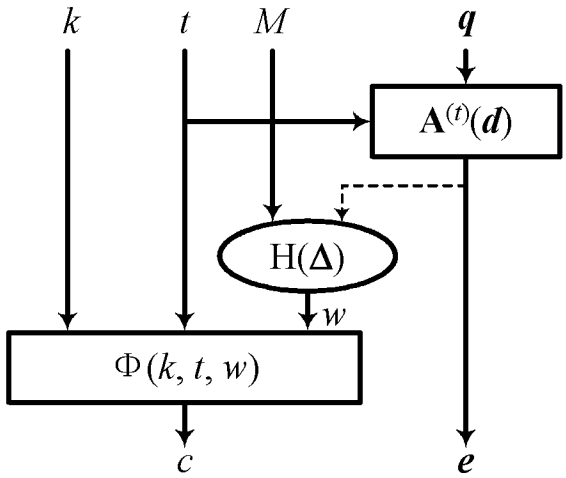
k
t
M
q
A(t)(d)
H(Δ)
w
Φ(k, t, w)
c
e

图 6

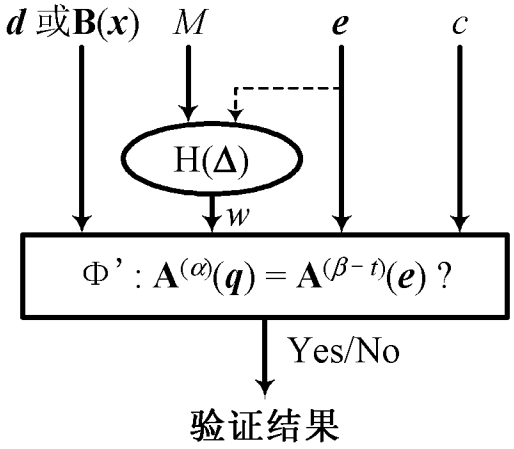
d 或B(x)
M
e
c
H(Δ)
w
Φ' : A^(α)(q) = A^(β−t)(e) ?
Yes/No
验证结果

图 7

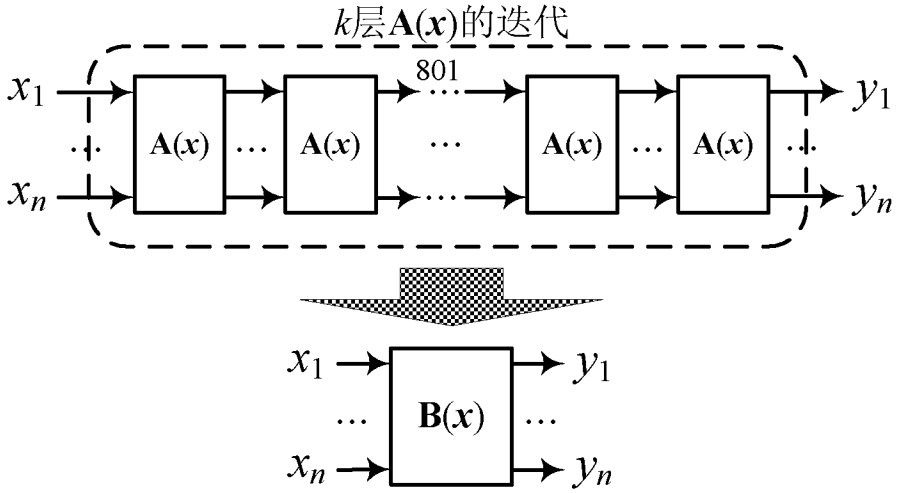
k层A(x)的迭代
801
x1
xn
A(x)
y1
yn
B(x)

图 8

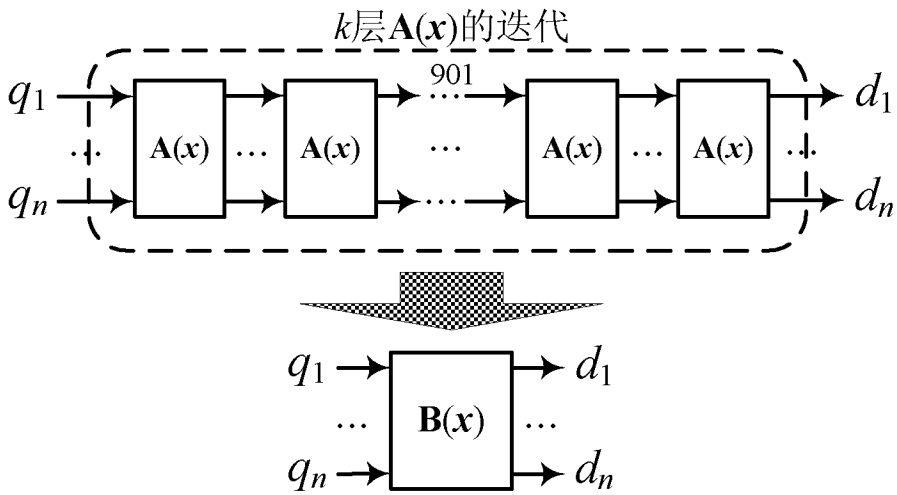
k层A(x)的迭代
901
q1
qn
A(x)
d1
dn
B(x)

图 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2007/070628

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L9/00,9/14,9/28,9/30,9/32,H04K1/00,1/02,G09C1/00,G06F7/72,7/38,7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI;EPODOC;PAJ;IEEE;CNPAT;CNKI: key, negotiat+, field, polynomial, rational fraction, rational function, signature, authenticat+, DH, ECC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GUAN HAIMING, Rational Function Public Key Cryptosystem, CCICS'2007, July 2007 (07.2007), page 139, lines 8-23, ISBN 978-7-03-019312-4	1-30
A	CN1831754A (BEIJING PUAODE DIGITIAL TECHNOLOGY CO., LTD.) 13 Sep. 2006 (13.09.2006) the whole document	1-30
A	US7096356B1 (Chen et al.) 22 Aug. 2006 (22.08.2006) the whole document	1-30
A	US2005/0149732A1 (Freeman et al.) 7 Jul. 2005 (07.07.2005) the whole document	1-30
A	JP2005284111A (DOKURITSU GYOSEI HOJIN KAGAKU GIJUTSU SH) 13 Oct. 2005 (13.10.2005) the whole document	1-30
A	US5375170A (Shamir) 20 Dec. 1994 (20.12.1994) the whole document	1-30

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&"document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 Apr. 2008 (08.04.2008)	**29 May 2008 (29.05.2008)**
Name and mailing address of the ISA/CN The State Intellectual Property Office, the P.R.China 6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088 Facsimile No. 86-10-62019451	Authorized officer LIU, Jianbo Telephone No. (86-10)62413304

Form PCT/ISA/210 (second sheet) (April 2007)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2007/070628

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN1831754A	13.09.2006	None	
US7096356B1	22.08.2006	None	
US2005/0149732A1	07.07.2005	None	
JP2005284111A	13.10.2005	None	
US5375170A	20.12.1994	EP0597481A2	18.05.1994
		EP0597481A3	05.04.1995
		EP0597481B1	17.10.2001
		DE69330934E	22.11.2001

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2007/070628

CLASSIFICATION OF SUBJECT MATTER

H04L9/32 (2006.01) i
H04K1/00 (2006.01) i

国际检索报告

国际申请号 **PCT/CN2007/070628**

A. 主题的分类

见附加页

按照国际专利分类表(IPC)或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC: H04L9/00,9/14,9/28,9/30,9/32,H04K1/00,1/02,G09C1/00,G06F7/72,7/38,7/00

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

WPI;EPODOC;PAJ;IEEE;CNPAT;CNKI：密钥，协商，域，多项式，有理分式，有理函数，签名，认证，key, negotiat+, field, polynomial, rational fraction, rational function, signature, authenticat+, DH, ECC

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	管海明,有理分式公钥密码体制, CCICS'2007, 7 月 2007 (10.2007), 第 139 页第 8-23 行, ISBN 978-7-03-019312-4	1-30
A	CN1831754A（北京浦奥得数码技术有限公司）13.9 月 2006（13.09.2006）参见全文	1-30
A	US7096356B1（Chen et al.）22.8 月 2006（22.08.2006）参见全文	1-30
A	US2005/0149732A1（Freeman et al.）7.7 月 2005（07.07.2005）参见全文	1-30
A	JP2005284111A（DOKURITSU GYOSEI HOJIN KAGAKU GIJUTSU SH）13.10 月 2005（13.10.2005）参见全文	1-30
A	US5375170A（Shamir）20.12 月 1994（20.12.1994）参见全文	1-30

☐ 其余文件在 C 栏的续页中列出。　　☒ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
8.4 月 2008（08.04.2008）	**29.5 月 2008 (29.05.2008)**
中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路 6 号 100088 传真号： (86-10)62019451	受权官员 刘剑波 电话号码：(86-10) **62413304**

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2007/070628

检索报告中引用的专利文件	公布日期	同族专利	公布日期
CN1831754A	13.09.2006	无	
US7096356B1	22.08.2006	无	
US2005/0149732A1	07.07.2005	无	
JP2005284111A	13.10.2005	无	
US5375170A	20.12.1994	EP0597481A2	18.05.1994
		EP0597481A3	05.04.1995
		EP0597481B1	17.10.2001
		DE69330934E	22.11.2001

国际检索报告

国际申请号
PCT/CN2007/070628

主题的分类

H04L9/32 (2006.01) i
H04K1/00 (2006.01) i