第八章、环签名及其在区块链中的应用

目　录

目　录

8.1 环签名的概念

1. 问题的提出

2001年，Rivest， Shamir和Tauman在*How to Leak a Secret*一文中提出了如下问题：

- Bob是一个内阁成员，他想向记者揭露首相贪污的情况，他要使记者确信此消息来自一个内阁成员，同时又不想泄露自己的身份（保证匿名性），以免遭到首相报复.
- Bob不能通过用一般的数字签名把消息传给记者，因为虽然记者会相信这个消息来自内阁成员，但同时会暴露B的身份；
- Bob也不能通过一般的匿名方式把消息传给记者，因为虽然Bob的身份不会暴露，但记者不能确信消息来自一个内阁成员，没有理由相信消息是真实的.
- 群签名也不能解决这个问题，因为群签名的生成需要群成员的合作， 群管理者可以打开签名.如果群管理者受到首相的控制， Bob的身份就会暴露.

8.1 环签名的概念

1. 问题的提出

在此背景下， Rivest， Shamir和Tauman提出了环签名的概念，并提出一种新型的基于RSA的环签名算法，通常被视为第一个环签名算法.

环签名可以很好的解决以上问题，所有的内阁成员构成一个环，Bob把消息经过环签名后传给记者， 在不暴露Bob身份的前提下记者可以通过验证环签名的正确性，确信签名来自一个内阁成员.

Rivest R L， Shamir A， Tauman Y . How to Leak a Secret[C]// Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. Springer， Berlin， Heidelberg，2001.

8.1 环签名的概念

2. 环签名的定义

环签名的概念是2001年Rivest， Shamir和Tauman三人提出的签名者模糊的数字签名.在环签名生成过程中，真正的签名者任意选取一组成员（包含它自身）作为可能的签名者，用自己的私钥和其他成员的公钥对文件进行签名.签名者选取的这组成员称作**环(Ring)**，生成的签名称作**环签名(Ring Signature)** .环签名主要包含以下三个算法：

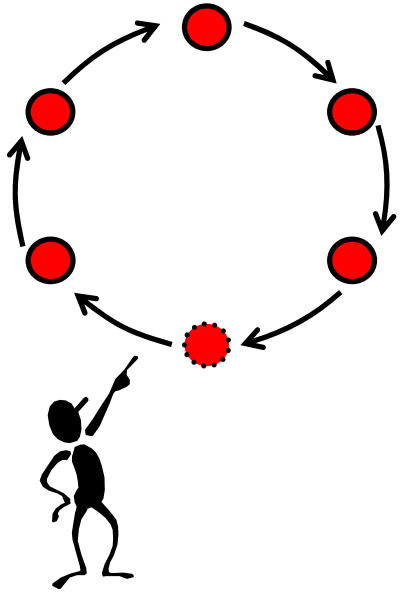
- **密钥生成Gen(PPT算法)**：输入安全参数κ，为每一用户u_i生成公私钥对(x_i ， y_i)，$1 \leq i \leq n$；
- **签名Sign (PPT算法)** ：输入消息m，一组公钥$L = \{y_1, y_2, \ldots, y_n\}$及签名者的私钥$x_s$，输出对$m$的签名$R$；
- **验证Verify (确定性算法)** ：输入(m, R)，输出“True”或“False”.

张国印， 王玲玲， 马春光. 环签名研究进展[J]. 通信学报， 2007(05):114-122.

8.1 环签名的概念

3. 环签名特点

- 签名者利用自己的私钥可以将签名中的一系列值首尾相连，环签名因其签名值由一定的规则组成一个环而得名；
- 没有群体建立过程，也无特殊的管理者；
- 不需要预先加入和撤出单个群体，群体的形成根据需要在签名前由签名人自己指定，是一种自组织结构；
- 不能追踪签名人身份，能通过验证确定签名者是其中某一人，但无人能指出具体是哪一位成员.



8.1 环签名的概念

4.环签名的安全性

- **正确性**：按照正确的签名步骤对消息进行签名，并且传播过程签名不被篡改，则环签名满足验证等式；
- **无条件匿名性**：攻击者即便非法获取了所有环成员的私钥，他能确定出真正签名者的概率不超过$1/r$，r是环中成员（可能的签名者）的个数；
- **不可伪造性**：环中其他成员不能伪造真实签名者的签名，攻击者即使在获得某个有效环签名的基础上也不能以不可忽略的优势成功伪造一个新消息的合法签名.

8.1 环签名的概念

5. 环签名与群签名比较

- **管理系统**：环签名中没有管理者，环中成员地位平等；群签名有一个群管理者，为群中其他成员分发密钥；
- **匿名性**：两者都是一种个体代表群体签名的体制，达到签名者匿名的作用.环签名具有无条件匿名性，任何人都不知道真正的签名者是谁；群签名中，签名可以由群管理者打开，确定真正的签名者，保证了签名的可追踪性；
- **组织结构**：环签名是一种自组织结构，真实签名者使用其他环成员的公钥时不需要他们的同意，环成员可以任意离开、加入；群签名中成员相对固定，群成员离开、加入群时密钥需要改变.

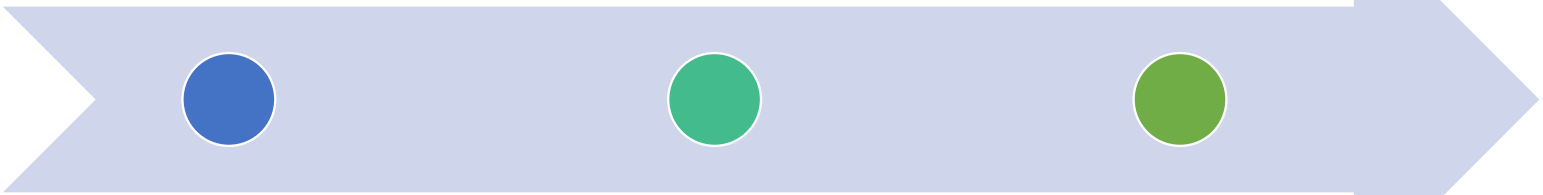
胡程瑜. (2008). *环签名体制的研究*. (Doctoral dissertation， 山东大学).

8.1 环签名的概念

6. 环签名的发展

环签名的发展经历了三个阶段：

2001-2002年，以Rivest等人提出的环签名定义为标志，这一阶段的工作主要参考Rivest等人的方案提出的签名方案；



2003-2004年，经过两年对环签名的概念和意义的认识和理解后，许多密码界人士开始对环签名进行深入研究，涌现了很多新思想、新模型和新方案，是环签名发展的关键时期；

2005年至今，这一阶段更加注重环签名的安全性、效率和实用性的研究，包括安全高效的环签名算法研究、环签名与通常的数字签名相互转化的研究以及环签名推广方面的研究.

张国印， 王玲玲， 马春光. 环签名研究进展[J]. 通信学报， 2007(05):114-122.

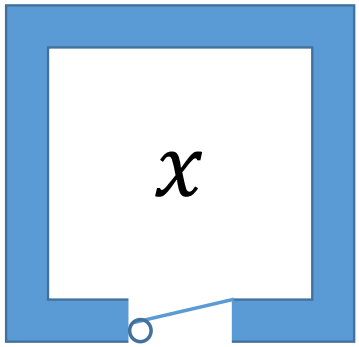
目　录

8.2 三个经典的环签名算法

1. RST环签名算法(第一个环签名算法)

首先给出单向陷门函数的定义：

- 给定x，可以较容易的计算$y=f(x)$，而给定y，计算满足$y=f(x)$的x即计算f^{-1}是困难的.
  但若得到对应的陷门t，计算f^{-1}是容易的.
- 单向陷门函数可以看作将给定的x放在一个有暗门的盒子当中，如下图：



单向陷门函数

[1]Rivest R L， Shamir A， Tauman Y . How to Leak a Secret[C]// Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. Springer， Berlin， Heidelberg，

8.2 三个经典的环签名算法

1. RST环签名算法

基于RSA算法的陷门函数：每个环成员$A_i(1 \le i \le r)$ 有一个RSA公私钥对 $(P_i = (e_i，\ n_i)$，$S_i = d_i)$，其中$e_i d_i = 1 \bmod \varphi(n_i)$，由此定义了单向陷门函数$f_i$：

$$f_i(x) = x^{e_i} (\mathbf{mod}\ n_i)$$

显然，只有拥有对应陷门信息d_i的环成员A_i能够计算出f_i^{-1}.

为了将单个签名方便的组合成环签名，将陷门函数f_i扩展到$\{0，\ 1\}^b$上的g_i

$$g_i(m) = \begin{cases} q_i n_i + r_i, & (q_i + 1)n_i \le 2^b \\ m, & otherwise \end{cases}$$

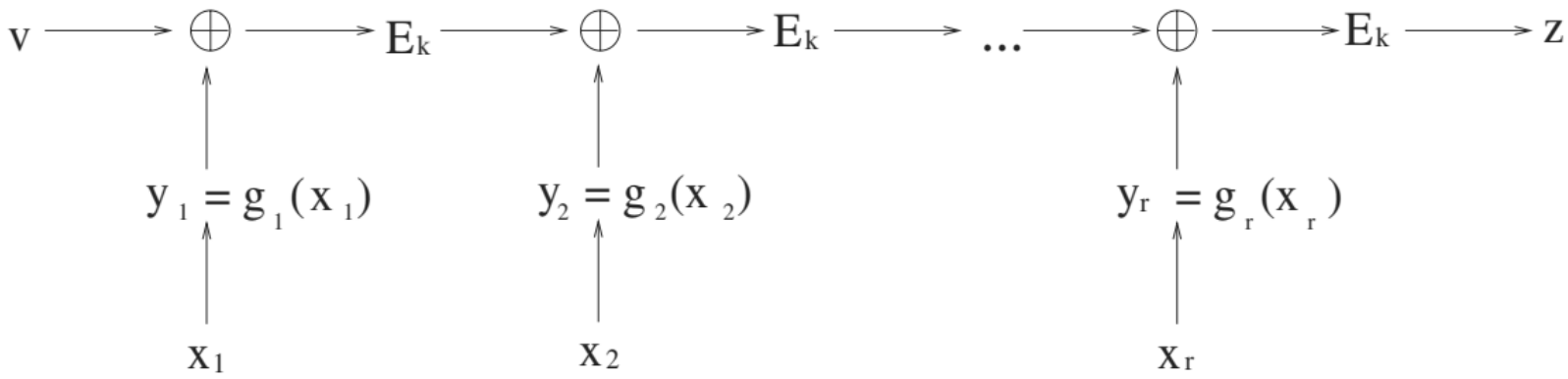
8.2 三个经典的环签名算法

1. RST环签名算法

组合函数：定义一种带密钥的组合函数$C_k，_v(y_1，y_2，…，y_r)$，输入为密钥k，初始值v和任意$y_1，y_2，…，y_r \in \{0，1\}^b$，以对称加密算法$E_k$为子程序，输出$z \in \{0，1\}^b$，计算方式及示意图如下

$$C_{k,v}(y_1, y_2, \ldots, y_r) = E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(y_{r-2} \oplus E_k(\ldots \oplus E_k(y_1 \oplus v)\ldots))))$$

$z =$



组合函数示意图

8.2 三个经典的环签名算法

1. RST环签名算法

对任意给定的密钥k和初始值v，组合函数$C_{k,v}(y_1, y_2, \ldots, y_r)$需满足以下三个性质：

① 对每个输入进行排列：对每个s，$1 \le s \le r$，固定$y_i(1 \le i \le r$且$i \ne s)$，组合函数为y_s到输出值z的一一映射；

② 单一输入有效可解：对每个s，$1 \le s \le r$，给定z和$y_i(1 \le i \le r$且$i \ne s)$，可有效解得y_s满足$C_{k,v}(y_1, y_2, \ldots, y_r) = z$；

例如，思考$k = 3$的情况：$C_{3,v}(y_1, y_2, y_3) = E_k(y_3 \oplus E_k(y_2 \oplus E_k(y_1 \oplus v))) = z$，若给定$z$，$y_1$，$y_3$，求满足$C_{3,v}(y_1, y_2, y_3) = z$的$y_2$.**提示：已知值为 k，v，z，y_1，y_3**

8.2 三个经典的环签名算法

保证环签名的
不可伪造性

1. RST环签名算法

③没有陷门输入不可求：给定k，[illegible]和z，如果敌手不能求得任一陷门函数g_1，g_2，…，g_r的逆，则求解满足如下等式的x_1，x_2，…，x_r是不可行的

$$C_{k,\,v}\left(g_1(x_1),\ g_2(x_2),\ \ldots,\ g_r(x_r)\right)=z$$

注：只有三个性质都满足的函数才能作为组合函数用于环签名，如$C_{k,\,v}(y_1,\ y_2,\ \ldots,\ y_r)=y_1\oplus y_2\oplus\ldots\oplus y_r$满足①②但不满足③，因而不能用于环签名（构造$x_1$，$x_2$，…，$x_r$ 的方法详见参考文献[1]）.

8.2 三个经典的环签名算法

8.2.1 RST环签名算法

算法描述：

(1)密钥对生成

生成陷门函数，公布函数并保存陷门信息d_i. (即$e_i \cdot d_i = 1 \bmod n_i$)

(2)环签名生成

给定待签名消息m，签名者π密钥S_π和所有环成员公钥P_1，P_2，…，P_r，签名者通过以下步骤生成环签名.

①. 选取密钥k：计算$k = h(m)$（或$k = h(m, P_1, \ldots, P_r)$）；

②. 选取随机粘合值：随机选择初始（粘合）值$v \in \{0, 1\}^b$；

③. 选取随机值x_i：为所有其他环成员选择$x_i \in \{0, 1\}^b (1 \le i \le r, i \ne \pi)$，并计算

$$y_i = g_i(x_i) \quad (\text{即} y_i = x_i^{e_i} \bmod n_i)$$

8.2 三个经典的环签名算法

8.2.1 RST环签名算法

④. 求解y_π：根据$y_i(1 \le i \le r，\ i \ne \pi)$，求满足如下等式的$y_\pi$

$$C_{k，\ v}(y_1，\ y_2，\ \ldots，\ y_r) = v$$

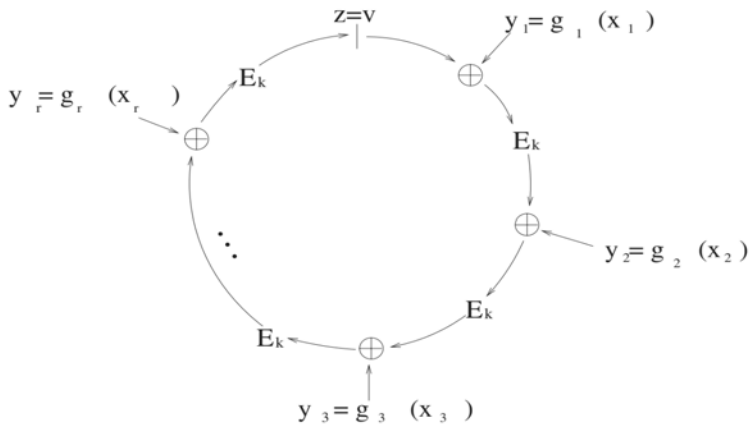
组合函数的性质②保证了方程可解；

⑤. 对陷门函数求逆：根据y_π和陷门信息（私钥）求逆

$$x_s = g_\pi^{-1}(y_\pi)\ (\text{即} x_\pi = y_\pi^{d_\pi} \bmod n_\pi)$$

⑥. 输出环签名：对m的环签名为一个$(2r+1)$元组

$$(P_1，\ \ldots，\ P_r; v; x_1，\ \ldots，\ x_r)$$



Rivest等人提出的环签名

8.2 三个经典的环签名算法

8.2.1 RST环签名算法

(3)环签名验证

根据对消息的m的签名$(P_1，\ \ldots，\ P_r; v; x_1，\ \ldots，\ x_r)$，验证者通过以下方式验证环签名.

①. 应用陷门函数：对$i = 1，\ 2，\ \ldots，\ r$，计算

$$y_i = g_i(x_i)$$

②. 获取密钥k：计算加密密钥k

$$k = h(m)\ （或k = h(m，\ P_1，\ \ldots，\ P_r)$$

③. 验证环等式：验证y_i是否满足

$$C_{k，\ v}(y_1，\ y_2，\ \ldots，\ y_r) = v$$

若等式成立，环签名为有效签名，否则为无效签名.

目　录

8.2 三个经典的环签名算法

8.2.2 AOS环签名算法

Abe等给出了一种通用环签名构造方法，利用该方法可以把任意数字签名转换为环签名，这里只介绍他们提出的基于Schnorr签名的环签名算法. 首先回顾一下Schnorr签名算法：

假设p和 q是大素数，是q能被p-1整除，q是大于等于160 bit的整数，p是大于等于512 bit的整数，保证GF(p)中求解离散对数困难；g是GF(p)中元素，且$g^q \equiv 1 \bmod p$.

➢ 密钥生成：

① Alice选择随机数x为私钥，其中$1<x<q$；

② Alice计算公钥$y \equiv g^x \pmod p$；

Abe M.， Ohkubo M.， Suzuki K. 1-out-of-n Signatures from a Variety of Keys. *ASIACRYPT 2002*. LNCS 2501， pp. 415–432， 2002.

8.2 三个经典的环签名算法

8.2.2 AOS环签名算法

- 签名算法

① Alice首先随机数k，这里$1<k<q$；

② Alice计算$c = h(M, g^k \bmod p)$

③ Alice计算$s = k - x \cdot c \ (\bmod q)$

④ Alice输出签名(c, s)

- 验证算法

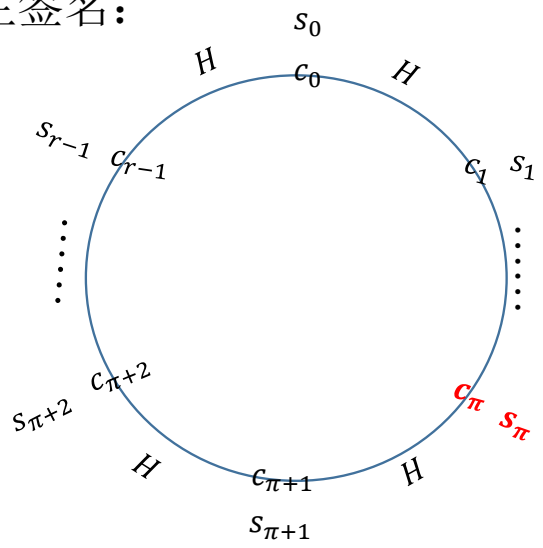
① Bob计算$e = g^s \cdot y^c \bmod p$.

② Bob验证$c = h(M, e)$是否成立，如果成立输出”Accept”，否则输出" Reject"

8.2 三个经典的环签名算法

8.2.2 AOS环签名算法

设第i个用户的私钥为x_i，公钥为$y_i = g^{x_i} \bmod p$. 签名者π的密钥对为(x_π, y_π)，随机生成公钥集$L=\{y_0, y_1, \ldots, y_{r-1}\}(0 \le \pi \le r-1)$，并执行以下过程产生签名：



➢ 环签名算法

① 选择随机数$k_\pi \in Z_q^*$并计算$c_{\pi+1} = H(L, M, g^{k_\pi} \bmod p)$；

② 选择随机数$s_i \in Z_q^*$并计算$c_{i+1} = H(L, M, g^{s_i} y_i^{c_i} \bmod p)$，

这里i=π+1，…，r-1，0，1，…，π-1.

③ 计算$s_\pi = k_\pi - x_\pi \cdot c_\pi \ (\bmod q)$

④ 输出环签名$(c_0, s_0, s_1, \ldots, s_{r-1})$

8.2 三个经典的环签名算法

8.2.2 AOS环签名算法

➢ 环验证算法

① 计算$e_i = g^{s_i} y_i^{c_i} \bmod p$和$c_{i+1} = H(L, M, e_i)$，这里$i$=0，1，…，$r$-1;

② 验证$c_0 = h(L, M, e_{r-1})$是否成立，如果成立则输出”Accept”，否则输出" Reject"

$$c_{\pi+1} = H\left(L, M, g^{s_\pi} y_\pi^{c_\pi} \bmod p\right)$$
$$= H(L, M, g^{k_\pi} \bmod p)$$

目　录

8.2 三个经典的环签名算法

8.2.3 ZK环签名算法

➢ 系统建立

令P是群G的生成元，阶为素数q.随机选取$s \in Z_q^*$作为TA主密钥，主公钥为$P_{pub} = s \cdot P$，哈希函数$H: \{0，1\}^* \to Z_q$，$H_1: \{0，1\}^* \to G$，双线性对$e: G \times G \to G_T$，系统参数$Params = \{G，q，P，P_{pub}，e，H，H_1\}$.

➢ 密钥提取

用户i的身份ID_i，私钥$S_{ID_i} = s \cdot H_1(ID_i)$，公钥$Q_{ID_i} = H_1(ID_i)$.

ZHANG F G， KIM K. ID-based blind signature and ring signature from pairings. *ASIACRYPT'02*. Springer-Verlag 2002.533-547

8.2 三个经典的环签名算法

8.2.3 ZK环签名算法

- 基于身份的签名算法

给定系统参数$Params$，消息m，私钥S_{ID_i}，签名者执行以下步骤产生签名：

① 随机选择$A \in G$，计算$c = H(m, e(A, P))$；

② 计算$T = A - c \cdot S_{ID_i}$；

③ 输出签名(c, T).

容易验证$e(A, P) = e(T, P)e(c \cdot H_1(ID_i), P_{pub})$
正确性成立！

- 基于身份的签名算法

给定系统参数$Params$，消息m，签名(c, T)，签名者执行以下步骤产生签名：

① 计算$c' = H(m, e(T, P)e(c \cdot H_1(ID_i), P_{pub}))$

② 判断c和c'是否相等.如果相等则输出”Accept”，否则输出" Reject“.

8.2 三个经典的环签名算法

8.2.3 ZK环签名算法

设第i个用户的身份为ID_i，私钥为S_{ID_i}. 签名者π随机生成身份集$L=\{ID_0, ID_1, \ldots, ID_{r-1}\}(0 \le \pi \le r-1)$，并执行以下过程产生签名：

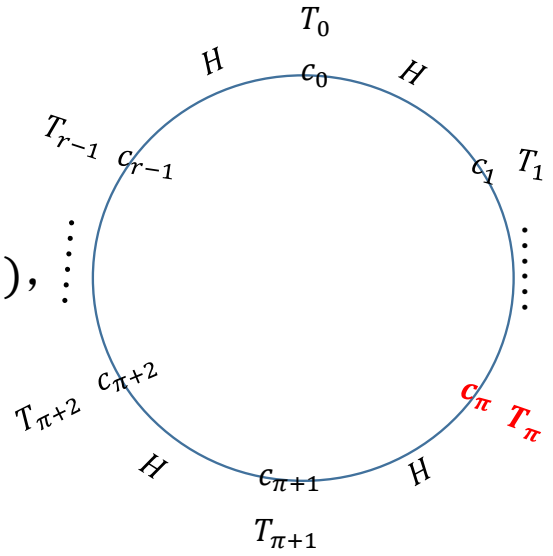
➢ 环签名算法

① 随机选择$A \in G$，计算$c_{\pi+1} = H(L||m||e(A, P))$；

② 随机选取$T_i \in G$，计算$c_{i+1} = H(L||m||e(T_i, P)e(c_i H_1(ID_i), P_{pub}))$，这里$i=\pi+1, \ldots, r-1, 0, 1, \ldots, \pi-1$；

③ 计算$T_\pi = A - c_\pi S_{ID_\pi}$；

④ 签名者输出环签名$(c_0, T_0, T_1, \ldots, T_{r-1})$.



8.2 三个经典的环签名算法

8.2.3 ZK环签名算法

➢ 环验证算法

给定$(c_0，\ T_0，\ T_1，\ \ldots，\ T_{n-1})$，$m$和$L$，

$i=0，\ 1，\ \ldots，\ r-1$，依次计算$c_{i+1}=H(L||m||e(T_i，\ P)e(c_iH_1(ID_i)，\ P_{pub}))$，若$c_r=c_0$验证通过，否则验证失败

$$
\begin{aligned}
&c_{\pi+1}\\
&=H\left(L||m||e(T_\pi，\ P)e\left(c_\pi H_1(ID_\pi)，\ P_{pub}\right)\right)\\
&=H\left(L||m||e(A，\ P)\right)
\end{aligned}
$$

目　录

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

1. SM2签名算法回顾

(1). 曲线参数

SM2标准推荐使用**256位素域F_p上的椭圆曲线$y^2=x^3+ax+b$**，其中：

p=FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF

a=FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFC

b=28E9FA9E 9D9F5E34 4D5A9E4B CF6509A7 F39789F5 15AB8F92 DDBCBD41 4D940E93

n=FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B 21C6052B 53BBF409 39D54123

G_x=32C4AE2C 1F198119 5F990446 6A39C994 8FE30BBF F2660BE1 715A4589 334C74C7

G_y=BC3736A2 F4F6779C 59BDCEE3 6B692153 D0A9877C C62A4740 02DF32E5 2139F0A0

(2). 密钥生成算法

① Alice选择随机数d_A做为私钥，其中0< d< n

② Alice计算公钥$P_A = dA \cdot G$

③ 输出密钥对 (sk=d_A， pk= P_A)

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

(3). 签名算法

- 设Alice发签名消息M给Bob，ID_A是Alice的标识符，$ENTL_A$是ID_A的长度，d_A是A的私钥，基点$G=(x_G, y_G)$，A的公钥$P_A=d_A \cdot G=(x_A, y_A)$.
- $Z_A=H(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$， H是SM3算法

① 设置$M^*=Z_A \| M$并计算 $e=H(M^*)$;

② 产生随机数 $k \in [1, n-1]$，计算椭圆曲线点 $G_1=k \cdot G=(x_1, y_1)$;

③ 计算 $c=(e+x_1) \bmod n$，若$r=0$或$r+k=n$则返回②； （修改为$c=H(M^*, G_1)$）

④ 计算 $s=(1+d_A)^{-1} \cdot (k-r \cdot d_A) \bmod n$，若$s=0$则返回②；

⑤ 以(c, s)作为对消息M的签名.

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

(4). 验证算法

接收到的消息为M'，签名为$(c'，\ s')$和发送者Alice的公钥P_A，Bob执行如下步骤验证合法性：

① 检验 $c' \in [1，\ n\text{-}1]$是否成立，若不成立则验证不通过；

② 检验 $s' \in [1，\ n\text{-}1]$是否成立，若不成立则验证不通过；

③ 设置$M^*=Z_A \| M'$ 并计算 $e'=H(M^*)$；

④ 计算 $t'=(c'+s') \bmod n$，若$t=0$，则验证不通过；

⑤ 计算椭圆曲线点 $G'_1=(x_1'，\ y_1')=s' \cdot G+(c'+s') \cdot P_A$；

⑥ 计算 $v=(e'+x_1') \bmod n$，　　(修改为$v=H(M^*，\ G'_1)$)

检验$v=r'$是否成立，若成立则验证通过；否则验证不通过.

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

2. 基于SM2签名算法的环签名

➢ 环签名算法

环签名群组公钥为$L=\{P_0，P_1，\ \ldots，P_{r-1}\}$，签名者为环成员之一，私钥$d_\pi$，公钥$P_\pi=d_\pi\cdot G$，待签名消息为$M$，$M^*=Z_A\|M$，$e=H(M^*)$.

1).随机选取$k_\pi\in\mathbb{Z}_n^*$，计算$Z_\pi=k_\pi\cdot G=(x_\pi，y_\pi)$，$c_{\pi+1}=x_\pi+e \bmod n$ (或者$c_{\pi+1}=H_1(L，M，k_\pi\cdot G)$)

2). 对$i=\pi+1，\ \ldots，r-1，0，1，\ \ldots，\pi-1$，随机产生$s_i\in\mathbb{Z}_n^*$，并依次计算

$$Z_i=s_i\cdot G+(s_i+c_i)\cdot P_i=(x_i，y_i)，\ c_{i+1}=x_i+e \bmod n(\text{或者}c_{i+1}=H_1(L，M，Z_i))$$

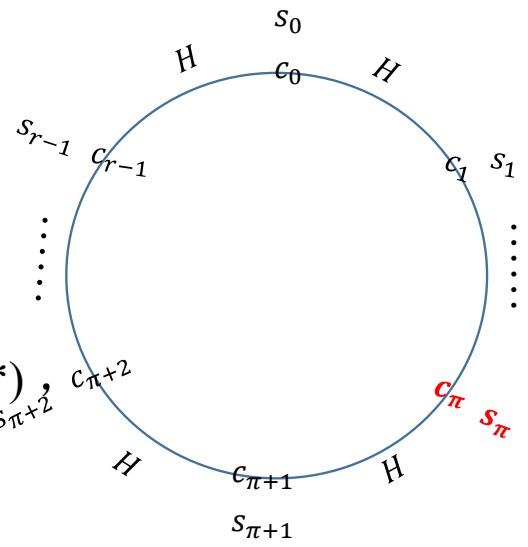
何德彪， 范青，彭聪，贾小英，罗敏，黄欣沂，一种基于SM2数字签名算法的环签名生成方法，申请公布号：CN110932866A

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

3). 计算$s_\pi = \left((1+d_\pi)^{-1} \cdot (k_\pi - c_\pi \cdot d_\pi)\right) \bmod n$

4). 输出环签名$\sigma_L(M) = (c_0,\ s_0,\ \ldots,\ s_{r-1})$



➢ 验证算法

收到$(M',\ c_0',\ s_0',\ \ldots,\ s_{r-1}')$后，设置$M^*=Z_A \| M'$，计算 $e'= H(M^*)$，通过以下步骤进行验证：

1). 检查$c_0',\ s_0',\ \ldots,\ s_r' \in \mathbb{Z}_n^*$是否满足，若不满足验证不通过；

2). 对$i = 0,\ 1,\ \ldots,\ r-1$，依次计算$Z_i' = s_i' \cdot G + (s_i' + c_i') \cdot P_i = (x'_i,\ y'_i)$，$c_{i+1}' = x'_i + e' \bmod n$；(或者$c'_{i+1} = H_1(L,\ M,\ Z_i')$)

3). 验证$c_r' \stackrel{?}{=} c_0'$，若成立则验证通过，否则验证不通过.

何德彪， 范青，彭聪，贾小英，罗敏，黄欣沂，一种基于SM2数字签名算法的环签名生成方法，申请公布号：CN110932866A

8. 3基于国密算法的环签名

8.3.1基于SM2签名算法的环签名

➢ 安全性：

- ✓ 正确性
- ✓ 无条件匿名性
- ✓ 不可伪造性

➢ 效率：

SM2环签名		
环签名	点乘运算	2*r+1
	哈希运算	r
环验证	点乘运算	2*r
	哈希运算	r

何德彪， 范青，彭聪，贾小英，罗敏，黄欣沂，一种基于SM2数字签名算法的环签名生成方法，申请公布号：CN110932866A

目　录

8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

1. SM9签名算法

➢统参数生成

密钥生成中心(Key Generation Center， KGC)执行以下步骤生成系统参数和主私钥：

① KGC生成随机数sk做为主私钥，这里0< sk <q-1；

② KGC计算系统公钥$P_{pub}=sk \cdot P_2$；

③ KGC保存私钥sk， 公布系统公钥.

- 注意：

① SM9算使用BN曲线，G_1和 G_2分别是椭圆曲线$E(F_p)$和$E(F_{p^2})$的加法群， G_T是乘法群 $F_{p^{12}}$ ，群G_2中元素尺寸是群G_1中元素尺寸的2倍.

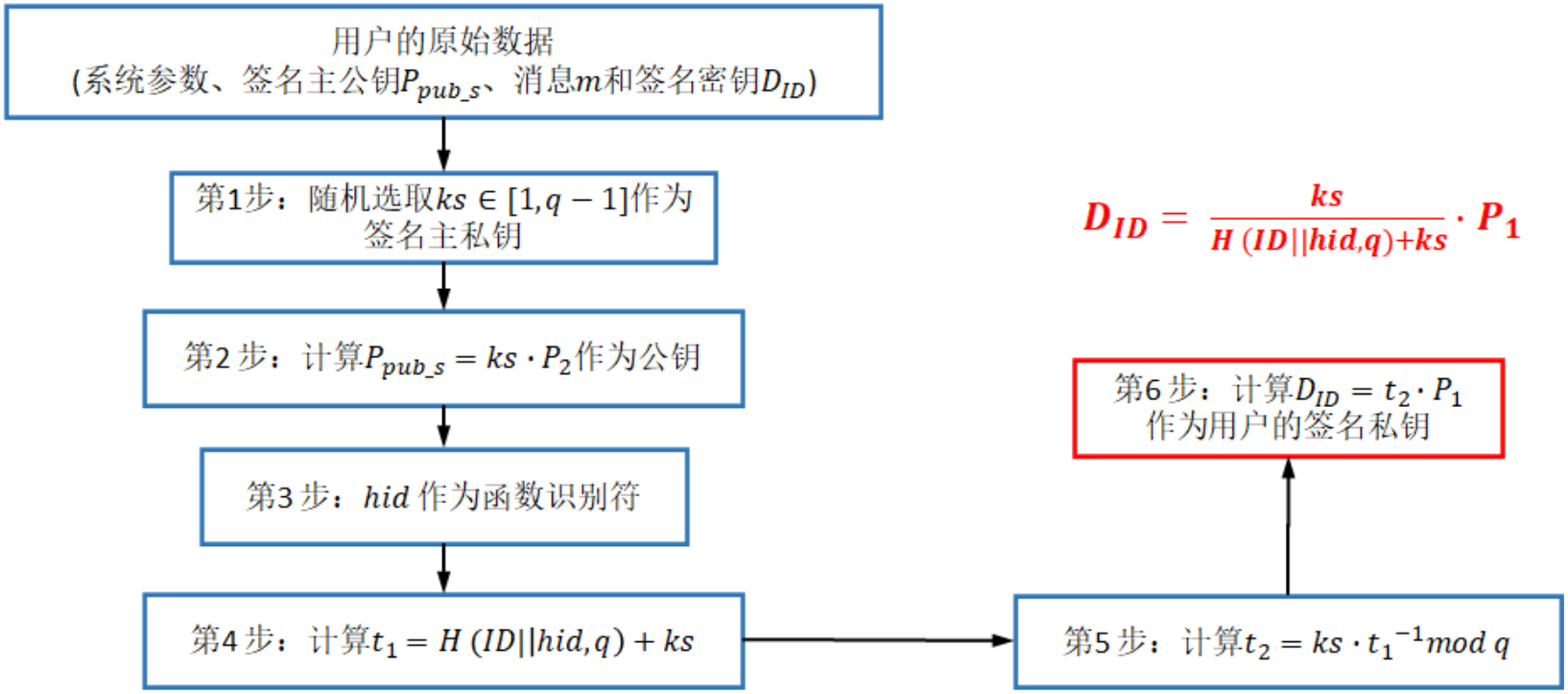
② 选择系统公钥为G_2中的元素，那么就可以使得用户私钥和签名中一部分是G_1中元素，降低了用户私钥和签名的尺寸.

8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

1. SM9签名算法

➢用户私钥生成

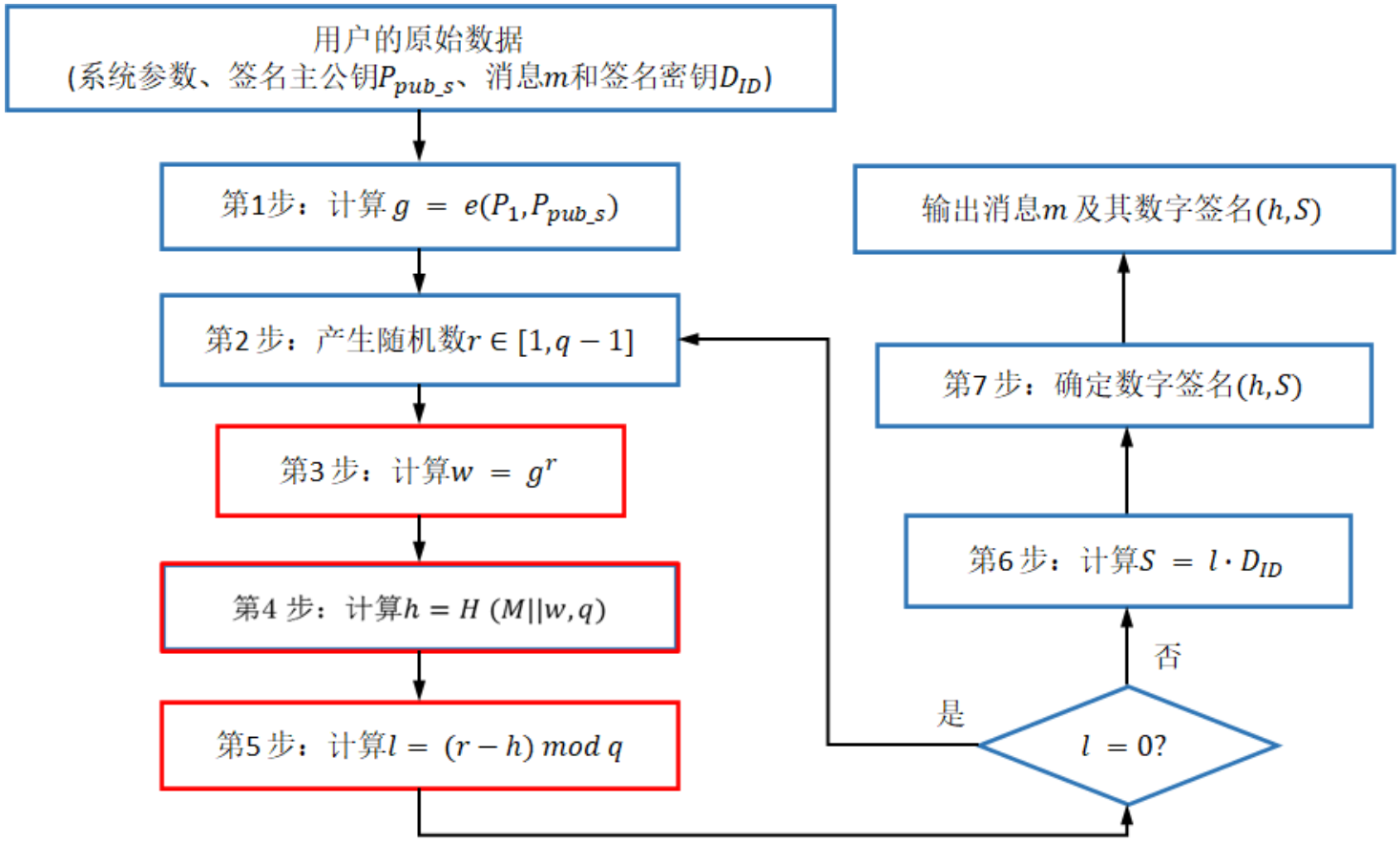


8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

1. SM9签名算法

➢签名算法

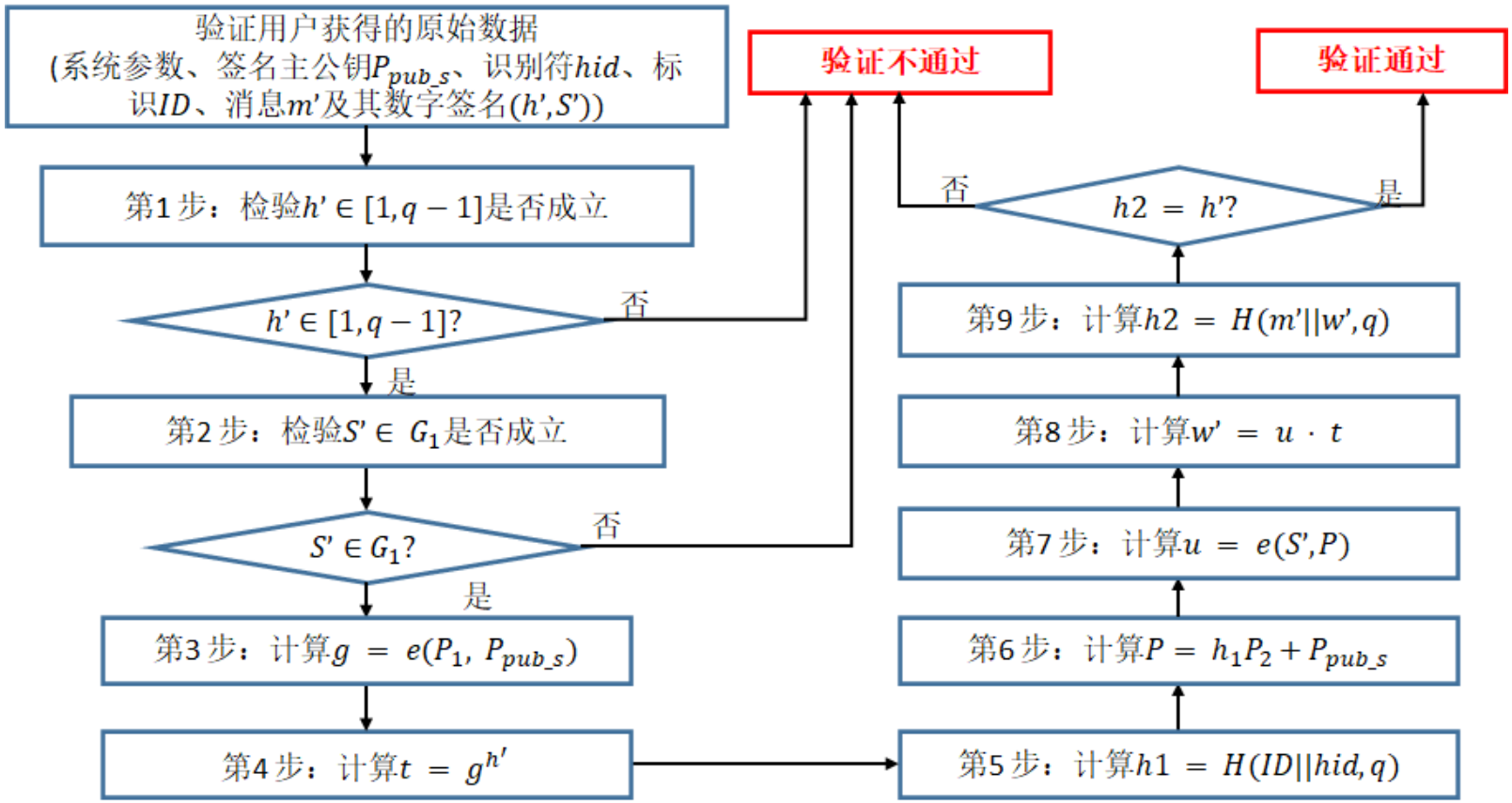


8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

1. SM9签名算法

➢ 验证算法

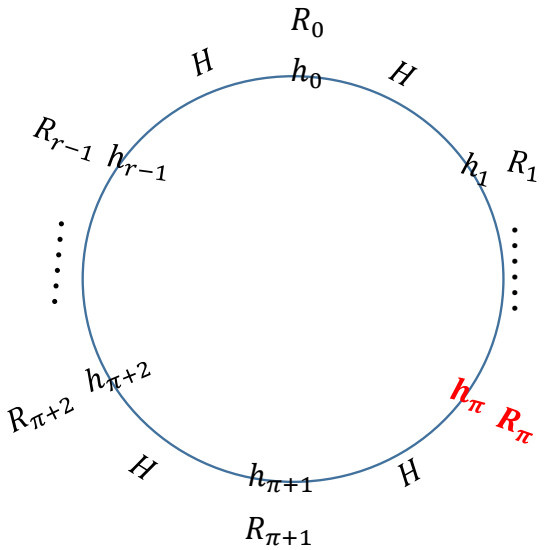


8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

2. 环签名算法

签名者标识为ID_π，私钥$D_{ID_\pi} = (s\,(H_1(ID_\pi||hid,\ N) + s)^{-1}) \cdot P_1$，给定消息$M$，签名者执行如下算法产生环签名：



① 签名者随机生成标识集合$L = \{ID_0,\ \ldots,\ ID_{r-1}\}$，$0 \le \pi \le r-1$；

② 随机选取$r_\pi \in Z_N^*$，计算$R_\pi = r_\pi \cdot P_1$，$w_\pi = e(R_\pi,\ P_{pub})$和$h_{\pi+1} = H_2(L||M||w_\pi,\ N)$；

③ 随机选取$r_i \in Z_N^*$，计算$R_i = r_i \cdot P_1$，$v_i = H_1(ID_i||hid,\ N)$，$Q_i = v_i \cdot P_2 + P_{pub}$，$u_i = e(R_i,\ Q_i)$，$w_i = u_i \cdot g^{h_i}$，$h_{i+1} = H_2(L||M||w_i,\ N)$，这里$i = \pi+1,\ \ldots,\ r-1,\ 0,\ 1,\ \ldots,\ \pi-1$.

④ 计算$R_\pi = (r_\pi - h_\pi) \cdot D_{ID_\pi}$；

⑤ 输出消息M的环签名$\sigma = (h_0,\ R_0,\ \ldots,\ R_{r-1})$

8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

3. 环验证算法

为了验证对消息M'的数字签名$\sigma' = (h_0'，R_0'，\ \ldots，R_{r-1}')$，验证者实现以下步骤.

① 计算$g = e(P_1，P_{pub})$;

② 对$i = 1，2，\ \ldots，n$，检验$h_i' \in Z_N^*$，$R_i' \in G_i$是否成立，并依次计算

$$v_i = H_1(ID_i||hid，N)，\ Q_i = v_i \cdot P_2 + P_{pub}，\ u_i' = e(R_i'，Q_i)$$

$$w_{i+1}' = u_i' \cdot g^{h_i'}，\ h_{i+1}' = H_2(L||M'||w_{i+1}'，N)$$

③ 验证$h_r' = h_0'$是否成立，若成立则验证通过，否则验证不通过.

8. 3基于国密算法的环签名

8.3.2基于SM9签名算法的环签名

4. 安全性与性能分析

➢ 安全性分析

- ✓正确性
- ✓不可伪造性
- ✓匿名性

➢ 性能分析

SM9环签名		方案一
环签名	点乘运算	*2n*
	双线性对运算	*n*
	G_T上乘运算	*n-1*
	G_T上幂运算	*n-1*
环验证	点乘运算	*n*
	双线性对运算	*n+1*
	G_T上乘运算	*n*
	G_T上幂运算	*n*

目　录

8. 4. 环签名在区块链中的应用

2013 年，Saberhagen 等人基于环签名技术提出了CryptoNote 协议，该协议提出了匿名电子现金系统在隐私方面需要满足的两大特性，并针对这两大特性设计协议.

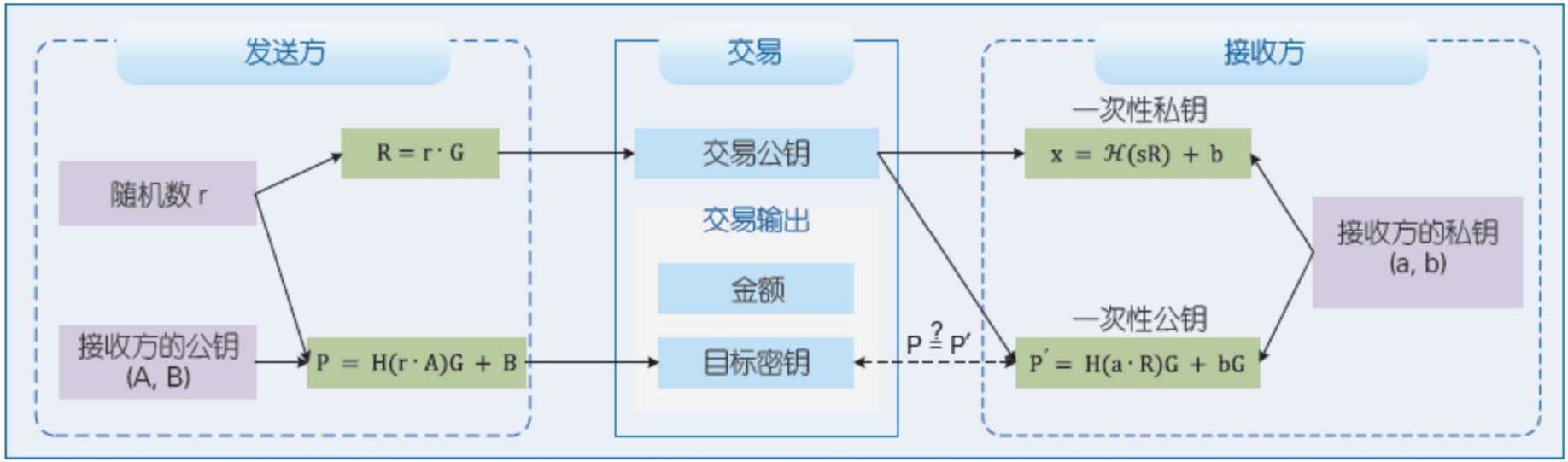
- 不可追踪性：对每个交易输入，所有可能的交易发起者都是等可能的;
- 不可链接性:对任意两个交易输出，不能证明他们是否发送给同一用户.

为了满足不可链接性，发送给同一接受者的不同资产需要发送到不同的地址，在传统区块链系统中，这需要接受者每次都生成新地址并从私密通道传递给发送者.在实际应用中，这给交易双方都带来不便.

为了解决这一问题，CryptoNote 采用了一次性公私钥对的方式，发起者可以根据接受者的长期公钥生成新的一次性公钥，新公钥只有接受者能计算出对应的私钥，并且不能被其他用户关联到接受者的长期公钥.这样，在保证了同一接受者每次交易存在不同接受地址的同时，发送者不需要接受者告知新公钥，可以独立构造交易.

van Saberhagen N. Cryptonote v 2.0， 2013[OL]//URL: https://cryptonote. org/whitepaper. pdf. White Paper. Accessed，2018: 04-13.

8.4. 环签名在区块链中的应用



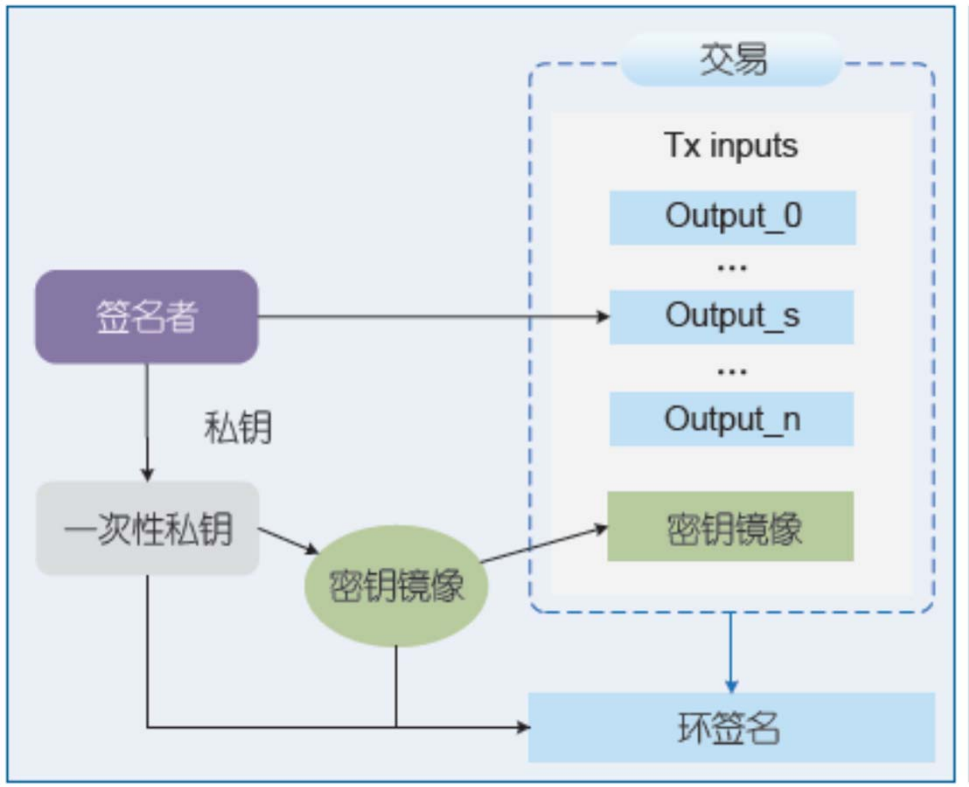
采用一次性公私钥对完成交易的流程如上图，分为交易发起者生成交易与接受者检查交易两部分.

- 交易发起者生成随机数与接受者公钥计算一次性公钥，作为交易的目标地址，并在交易中公布随机数对应的交易公钥.
- 交易接受者用私钥与交易公钥计算一次性公钥判断是否为交易目标地址:如果匹配，则能用私钥对该输出资产进行签名并花费.

8.4. 环签名在区块链中的应用

为了满足不可追踪性，在发起交易的时候，CryptoNote协议在一次性公私钥对的基础上采用一次性环签名技术保护交易发送者隐私.

一次性环签名同样分为生成签名与验证签名两阶段，并利用私钥哈希值作为快照保证资产对应私钥只进行过一次花费. 交易赎回后即丢弃，保证交易目标地址不会重复。



8. 4. 环签名在区块链中的应用

CryptoNote 协议在交易中采用一次性公私钥对保护了接收方的隐私，采用一次性环签名保护了发送方的隐私.但CryptoNote 协议采用环签名保护用户隐私的主要问题在于用户隐私依赖于选择的公钥集合，一旦集合中的其他用户公开使用了资产或发生隐私泄漏，导致输出被关联到对应地址，则该用户的地址关联情况也会被攻击者分析得出，导致隐私泄漏.

Noether S， Noether S， Mackenzie A. A note on chain reactions in traceability in cryptonote 2.0. Research Bulletin MRL0001. Monero Research Lab， 2014，12(1):1-8.

8. 4. 环签名在区块链中的应用

此后，ByteCoin[1]，Boolberry[2]，DigitalNote[3]等一系列区块链项目都采用了CryptoNote作为底层框架来保护用户隐私. 其中，应用较为广泛的是Surae等人提出的Monero项目[4]，该项目通过强制每笔环签名交易至少包含2个额外资产的方式，解决上述隐私泄漏的风险，增强了用户的长期匿名性[5].

[1]. BytecoinCommunity. Bytecoin project. https://bytecoin.org/

[2]. BoolberryTeam. Block chain based proofofwork Hash. https://boolberry.com/files/Block_Chain_Based_Proof_of_Work.pdf

[3]. DigitalNote project. https://www.digitalnote.biz/

[4]. Noether S. Review of crypto note white paper. https://downloads.getmonero.org/whitepaper_review.pdf

[5]. Noether S, Noether S. Monero is not that mysterious. https://web.getmonero.org/resources/researchlab/pubs/MRL0003.pdf

谢谢！