Advances in Applied Mathematics 应用数学进展, 2021, 10(6), 1880-1886
Published Online June 2021 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2021.106197

Hans汉斯

一种改进的椭圆曲线群签名方案

邱中保，孟忻怡，宫丹丹，卫培超，黄永清，张　平*

河南科技大学数学与统计学院，河南　洛阳
Email: *zhangping76@126.com

收稿日期：2021年5月8日；录用日期：2021年5月28日；发布日期：2021年6月9日

摘　要

为了实现群内成员的签名的公开性同时保证签名对群外用户的匿名性，本文提出了一种改进的群签名方案。该方案在椭圆曲线数字签名算法的安全性和高效性基础上，在群内引入两个主体对群进行管理和简化签名过程。算法安全性分析表明，该方案具有不可伪造性、抗密钥泄露、防数据篡改、防陷害性和不可抵赖性。算法效率分析表明，本方案与传统的椭圆曲线数字签名算法相比大大简化了运算复杂度，提高了算法的效率。相较于肖帅等人的方案，总体效率基本相同，但却能在验证方式相同情况下，根据对象所有的公钥类型不同，同时对两类对象实现效果不同的签名。

关键词

椭圆曲线，数字签名，求逆运算，匿名性

An Improved Group Signature Scheme for Elliptic Curves

Zhongbao Qiu, Xinyi Meng, Dandan Gong, Peichao Wei, Yongqing Huang, Ping Zhang*

School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang Henan
Email: *zhangping76@126.com

Received: May 8th, 2021; accepted: May 28th, 2021; published: Jun. 9th, 2021

Abstract

In order to realize the openness of group members' signatures and ensure anonymity of signa-

*通讯作者。

文章引用: 邱中保, 孟忻怡, 宫丹丹, 卫培超, 黄永清, 张平. 一种改进的椭圆曲线群签名方案[J]. 应用数学进展, 2021, 10(6): 1880-1886. DOI: 10.12677/aam.2021.106197

tures to users outside the group, an improved group signature scheme is proposed in this paper. Based on the security and efficiency of elliptic curve digital signature algorithm, two agents are introduced to manage the group and simplify the signature process. The algorithm security analysis shows that the scheme is unforgeable, anti-key leakage, anti data tampering, anti framing and non repudiation. The efficiency analysis shows that compared with the traditional elliptic curve digital signature algorithm, this scheme greatly simplifies the computational complexity and improves the efficiency of the algorithm. Compared with scheme of Xiao Shuai, the overall efficiency is basically the same, but under the same verification mode, according to the different public key types of the objects, the two kinds of objects can be signed with different effects.

Keywords

Elliptic Curve, Digital Signature, Inversion Operation, Anonymity

1. 引言

1985 年 Victor Miller [1]和 Neal Koblitz [2]分别提出了椭圆曲线公钥密码体制(Elliptic Curve Cryptography, ECC)，该体制的安全性是基于椭圆曲线离散对数求解的困难性。相比于其他密码体制，ECC 具有计算量小，存储消耗低等优点，椭圆曲线密码体制现已成为一种信息安全标准。基于椭圆曲线密码体制 Johnson 等人在 2001 年提出了椭圆曲线数字签名算法(Elliptic Curve Digital Signature, ECDSA)，2005 年 Brown [3]，给出了 ECDSA 方案的不可伪造性证明。后来又有多种改进方案被相继提出。

2008 年，潘晓君[4]提出了一种新的无模逆的签名算法。2009 年，侯爱琴等人[5]给出了一种高效的数字签名方案，该方案对 Hash 值的 Hamming 重量进行签名，提高了运算效率。2011 年，陈亮[6]等人对椭圆数字签名算法进一步优化与设计，新型的 ECDSA 签名方案在签名、验证过程中不仅避免了模逆运算，而且减少了点乘运算。同年，许德武等人[7]将 ELGamal 签名方案与椭圆曲线体制的安全性相结合，使用代数运算代替椭圆曲线上的数乘运算，进一步提高系统的安全性。2014 年，王国才等人[8]提出了一种高效的分级群签名方案，该方案通过对椭圆曲线群签名方案进行改进，在签名算法与验证上避免了模逆和模乘运算，具有较强的实用性。2015 年，白永祥[9]基于椭圆曲线密码系统的优势设计了一种群签名方案，并且使用杂凑函数 SHA-3 提高了签名的安全性。2019 年，陈亚茹等人[10]提出了一种改进方案，该方案结合 Hamming 距离，通过两次点乘运算和一次求逆运算提高了数字签名的计算效率。2020 年，肖帅等人[11]设计出一种较为高效的方案，但该方案并未给出形式化的安全性证明。同年，张平等人[12]针对椭圆曲线数字签名方案中通过替换消息伪造签名的问题给出了一种解决方案，其安全性较经典方案有所提高。

Chaum 等人[13]提出群签名的概念，2020 年苏吟雪[14]等人提出了一种基于 SM2 的双方共同签名协议,该协议为了进一步保护用户信息安全，协议要求签名所用私钥的一部分存储在服务器中，虽然增加了服务器认证用户的机会，但是完成签名的步骤增加了，使签名的效率有所降低。本文沿用文献[11]的椭圆曲线数字签名算法，重点将双方共同签名协议拓展到多方，实现一种特殊类型的群签名方案。方案中群成员因内部消息不再具有匿名性，所以无需签名打开算法，其匿名性体现在对外的签名验证上。

2. 预备知识

2.1. 椭圆曲线离散对数问题

对于有限域 F_p (要求 p 为一个大素数)，定义在 F_p 上的椭圆曲线 E 为：$y^2 = x^3 + ax + b \pmod p$ 且 $4a^3 + 27b^2 \neq 0 \pmod p$。椭圆曲线离散对数的问题就是：若已知 G，Q 两点，寻找一个小于 p 的整数 d 使得 $Q = dG$ 是困难的，即无法在多项式时间内完成。

2.2. ECDSA 方案描述

2.2.1. 参数选择

设 ECC 的参数为 $T = (p, a, b, G, n)$，其中 p 为一个很大的素数，F_p 是一个有限域，G 是 F_p 上的 n 阶基点，n 为一个素数。S 为签名者，其公私钥对为 (Q_s, d_s)，V 为验证者。$H: \{0,1\}^* \to \{0,1\}^{n^*}$ 是一种安全 Hash 算法。

2.2.2. 签名算法

1) S 随机选择一个整数 $k\ (1 \le k \le n-1)$。
2) 计算 $kG = (x_1, y_1)$ 和 $r = x_1 \bmod n$；若 $r = 0$，则返回步骤 1。
3) 计算需要签名的消息 m 的哈希值 e，$e = H(m)$。
4) 计算 $s = k^{-1}(e + d_s r) \bmod n$，若 $s = 0$ 没，则返回步骤 1。
5) 输出签名 $\sigma = (s, r)$。

2.2.3. 验证算法

1) 验证 s，r 是否为区间 $[1, n-1]$ 内的整数，若验证失败，则拒绝签名。
2) 计算签名消息 m 的哈希值 e，$e = H(m)$。
3) 计算 $u_1 = es^{-1} \pmod n$，$u_2 = rs^{-1} \pmod n$。
4) 计算 $R = u_1 G + u_2 Q = (x_1, y_1)$，若 $R = 0$，则验证失败，拒绝签名。
5) 计算 $v = x_1 \bmod n$，若 $v = r$，则接受签名，否则拒绝签名。

3. 本文方案

设椭圆曲线的参数为 $T = (p, a, b, G, n)$，具体过程如下：

3.1. 用户密钥产生算法 Ukge

1) 用户 $U_i\ (i = 1, 2, \cdots, m)$ 随机选取一个整数 d_i，$1 < d_i \le n-1$ 作为私钥。
2) U_i 计算 $Q_i = d_i \cdot G$ 作为公钥。
3) 输出公私钥对 (Q_i, d_i)。

3.2. 群密钥产生算法 Gkge

在准备阶段，每个群成员公开自己的公钥 $Q_i\ (i = 1, 2, \cdots, m)$，群中心计算 $Q = Q_1 + Q_2 + \cdots + Q_m$ 作为群公钥，那么 $d = d_1 + d_2 + \cdots + d_m \pmod n$ 就成为群私钥，但群内任何一位成员都不知道 d 的具体数值。

3.3. 成员加入算法 Join

若用户 A 想成为群中的一个成员，首先 A 要向管理员提出申请，获得管理员授权后，A 运行 Ukge 算法，获得公私钥对 (Q_A, d_A)。然后用户 A 和管理员进行交互(在安全通道上进行)，用户 A 向管理员发送

自己的公钥信息，然后接收管理员所发送的群基本信息(包括群内成员，各个成员的公钥等)。A 和管理员交互之后，管理员将 A 的公钥在群内公开，之后 A 便成为一名合法的群成员。群中心更新群的公钥为 $Q'=Q+Q_A$。此时，群的私钥变为 $d'=d+d_A \pmod n$。从成员加入算法过程可以得出：当加入新成员时并不需要改变其他群成员的公私钥信息，只需重新计算群公钥 Q 即可。

3.4. 群签名生成算法 Gsig

假设用户 U_1 的公私钥对为 (Q_1,d_1)，签名过程如下：

1) U_1 随机选取两个整数 α,β，其中 $1\le\alpha$，$\beta\le n-1$。

2) 计算 $k=(\alpha d_1+\beta) \bmod n$；若 $k=0$，则返回步骤 1。

3) 计算 $kG=(x_1,y_1)$ 和 $r=x_1 \bmod n$；若 $r=0$，则返回步骤 1。

4) 计算需要签名的消息 m 的哈希值 e，$e=H(m)$。

5) 计算 $s=(\beta+d_1er) \bmod n$，若 $s=0$，则返回步骤 1。

6) 计算 $(\alpha-er)\cdot Q^{\sim}+kG=(x_2,y_2)$ (这里 $Q^{\sim}$ 是除去 U_1 后剩余群成员的公钥之和，$Q^{\sim}$ 是不需要多次计算的，只需计算一次后储存起来，之后直接使用即可)。

7) 计算 $r'=x_2 \bmod n$；若 $r'=0$，则返回步骤 1。

8) 输出签名 $\sigma=(s,\alpha,r,r')$。

注意到该算法的 6~8 步是针对于群成员对外的签名生成过程，若不考虑 $r'=0$ 的情况(可能性很小)，则可以将 6~8 步交由群中心完成，若签名不对外发放，则 6~8 步可以直接跳过，输出签名 (s,α,r) 即可。

3.5. 验证算法 GVer

根据验证对象使用公钥类型的不同可以分为两类：群内成员的验证和群外成员的验证。

3.5.1. 群内部成员的验证

对于群内部成员 B_1 来说，B_1 拥有 U_1 的公钥，验证 U_1 的签名时只需要 (s,α,r) 即可，验证步骤如下：

验证 s,α,r 是否为区间 $[1,n-1]$ 内的整数，若验证失败，则拒绝签名。

1) 计算消息 m 的哈希值 e，$e=H(m)$。

2) 计算 $u=er$。

3) 计算 $sG+(\alpha-u)Q_1=(x_3,y_3)$。

4) 计算 $v=x_3 \bmod n$。

5) 验证 v 和 r 的关系，若 $v=r$，则接受签名，否则拒绝签名。

正确性分析如下：

若 s,α,r 是 A_1 对消息 A_1 的签名信息，则：

$$(x_3,y_3)=sG+(\alpha-u)Q_1=(\beta+d_1er)G+(\alpha-er)d_1G=(\alpha d_1+\beta)G=kG=(x_1,y_1)$$

所以有：$v=x_3=x_1=r \bmod n$。

3.5.2. 群外部成员的验证

对于群外部成员 B_2 来说，B_2 只有群的公钥，验证步骤如下：

1) 验证 s,α,r,r' 是否为区间 $[1,n-1]$ 内的整数，若验证失败，则拒绝签名。

2) 计算消息 m 的哈希值 e，$e=H(m)$。

3) 计算 $u=er$。

4) 计算 $sG+(\alpha-u)Q_1=(x_4,y_4)$。

5) 计算 $v = x_4 \bmod n$ 。

6) 验证 v 和 r' 的关系，若 $v = r'$ ，则接受签名，否则拒绝签名。

正确性分析如下：

若 (s, α, r, r') 是群对消息 m 的签名信息，则：

$$\begin{aligned}(x_4, y_4) &= sG + (\alpha - u)Q_1 \\ &= (\beta + d_1 er)G + (\alpha - er)(d_1 + d_2 + \cdots + d_m)G \\ &= kG + (\alpha - er)(d_1 + d_2 + \cdots + d_m)G \\ &= kG + (\alpha - er)\widetilde{Q} \\ &= (x_2, y_2)\end{aligned}$$

所以有： $v = x_4 = x_2 = r' \bmod n$ 。

4. 安全性分析

1) 不可伪造性：

若攻击者能够替换消息，虽然能够计算出替换后消息的哈希值，但会话密钥和签名者的私钥都是未知的，不能够伪造出可以通过验证的签名。

若攻击者能够用随机数 k' 来替换 k，但接收方验证出来的 v 和 r (或 r')不同，所以可以有效地抵御伪造攻击。

2) 抗密钥泄露：

因为大多数情况下，一旦随机数相同就可以构造出一个二阶的线性方程组解出用户的私钥，从而造成私钥的泄露，所以一般来说对不同的消息使用同一签名方案进行签名时，使用的随机数是不相同的。如果本文方案对不同消息进行签名时都使用相同的随机数，那么就可以根据方程组：

$$\begin{cases} s_1 = (\beta + d_1 e_1 r_1) \bmod n \\ s_2 = (\beta + d_1 e_2 r_2) \bmod n \end{cases}$$

(其中 $e_1, s_1, e_2, s_2, r_1, r_2$ 是已知的)， β, d_1 是未知的，得出私钥的表达式为

$$d_1 = (r_2 - r_1)^{-1}(e_2 - e_1)^{-1}(s_2 - s_1)^{-1} \bmod n$$

如果每次签名的随机数不同，想通过上述攻击方法来破解该方案是无法实现的。假设 β 随机后，上述方程组就变为

$$\begin{cases} s_1 = (\beta_1 + d_1 e_1 r_1) \bmod n \\ s_2 = (\beta_2 + d_1 e_2 r_2) \bmod n \end{cases}$$

未知量为 β_1, β_2, d_1 。未知量的个数就由两个变为三个，根据代数知识，不能求出私钥。根据椭圆曲线离散对数难题，已知方程组：

$$\begin{cases} (\alpha_1 - e_1 r_1)\widetilde{Q} + k_1 G = (x_{21}, y_{21}) \\ (\alpha_2 - e_2 r_2)\widetilde{Q} + k_2 G = (x_{22}, y_{22}) \end{cases}$$

也是无法得到 $\widetilde{Q}$ 的。所以本文方案在保证随机数不同时，能够防止密钥泄露。

3) 防数据篡改：

在产生签名时需要待签名信息的哈希值参与，数据的完整性得到了保证，一旦信息发生变化，其对

应的哈希值也发生变化，从上述签名信息 s 的计算方式可以看出，一旦待签名信息哈希值 e 发生变化，群内部和外部成员都无法通过签名验证，从而保证了签名数据不会被篡改。

4) 防陷害性：

由于群内部成员的签名需要各自的私钥参与运算，而用户的私钥是不公开的，因此任何成员都不能以其他成员的名义对群内部进行签名。

5) 不可抵赖性：

接收方可以根据发送方的消息签名 $\sigma=(s,\alpha,r,r')$ 来防止发送方的抵赖。

5. 效率分析

从算法过程可以看出，耗时计算主要集中在签名和验证的执行上。由于加法的运算量较小对算法整体的影响不大，暂且忽略不计。设模乘运算的数据规模是 n，一次点乘运算的复杂度为 $O(n^2)$，一次求逆运算复杂度大约是点乘运算的 9 倍，即 $O(9n^2)$。根据文献[11]一次模乘运算的复杂度为 $O(n^2 lnn)$。将本文方案与 ECDSA 方案和肖帅等人方案的运算量进行对比结果见表 1。

Table 1. Comparison of computation cost of three schemes
表 1. 三种方案运算量对比

方案	模乘运算		求逆运算		点乘运算	
	签名	验证	签名	验证	签名	验证
ECDSA	2	2	1	1	1	2
肖帅等人的方案	3	2	0	0	1	1
本文方案(仅对内签名)	3	1	0	0	1	2
本文方案(同时对内外签名)	3	1	0	0	2	2

由表 1 可知，ECDSA 的总运算量为：$N_1=O(4n^2 lnn+21n^2)$。

肖帅等人方案的总运算量为：$N_2=O(5n^2 lnn+2n^2)$。

本文方案(仅对内签名)的总运算量为：$N_3=O(4n^2 lnn+3n^2)$。

本文方案(同时对内外签名)的总运算量为：$N_4=O(4n^2 lnn+4n^2)$。

本文方案在对内的签名和验证上总体效率优于 ECDSA 方案，和肖帅等人的方案效率近似。在签名阶段，本文方案比 ECDSA 方案多了 1 次乘法运算，少了 1 次模逆运算；在验证阶段，本文方案比 ECDSA 方案少了 1 次模乘运算，总体上避免了求逆运算，提高了算法的效率。在对外的签名和验证上多了一次点乘运算，但是本文方案给出实现了新的功能，并且可以让群内外验证人员执行相同的验证步骤。总体来说，本文方案加强了算法的安全性，提高了算法的实用性。

6. 结语

本文首先对于 ECDSAC 密码体制进行分析，发现该方案 ECC 具有计算量小，处理速度快等优点。又分析了传统群签名在某些需要公开签名的应用场景下管理员需要反复执行签名打开算法的问题。本文提出了一种既能实现群内成员签名的公开性，又保证签名对群外用户匿名性的签名方案。并对该方案进行安全性和效率分析，发现本方案在保证安全性的情况下还具有较高的效率。在一个不需要经常撤销群内成员的环境下具有很强的适用性，能够很好地满足某些群的公开性需求。

基金项目

河南省高等学校重点科研项目(项目编号：20A520012)；河南科技大学大学生研究训练计划(SRTP)项目(项目编号：2020173)。

参考文献

[1] Miller, V. (1986) Uses of Elliptic Curves in Cryptography. LNCS 218: Advances in Cryptology. Springer, Berlin, 387-398.

[2] Koblitz, N. (1987) Elliptic Curve Cryptosystem. *Mathematics of Computation*, **48**, 203-209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

[3] Brown, D.R.L. (2005) Generic Groups, Collision Resistance, and ECDSA. *Designs, Codes and Cryptography*, **35**, 119-152. https://doi.org/10.1007/s10623-003-6154-z

[4] 潘晓君. 一种新的基于椭圆曲线的数字签名方案[J]. 计算机系统应用, 2008(1): 35-37.

[5] 侯爱琴, 高宝建, 张万绪, 等. 基于椭圆曲线的一种高效率数字签名[J]. 计算机应用与软件, 2009, 26(2): 58-69+71.

[6] 陈亮, 游林. 椭圆曲线数字签名算法优化与设计[J]. 电子器件, 2011, 34(1): 89-93.

[7] 许德武, 陈伟. 基于椭圆曲线的数字签名和加密算法[J]. 计算机工程, 2011, 37(4): 168-169+189.

[8] 王国才, 刘美兰. 基于椭圆曲线的高效分级群签名[J]. 计算机应用研究, 2014, 31(2): 586-589.

[9] 白永祥. 一种群签名方案的设计与分析[J]. 智能计算机与应用, 2015, 5(1): 14-17.

[10] 陈亚茹, 丛培强, 陈庄. 一种椭圆曲线数字签名的改进方案[J]. 信息安全研究, 2019, 5(3): 217-222.

[11] 肖帅, 王绪安, 潘峰. 无模逆运算的椭圆曲线数字签名算法[J]. 计算机工程与应用, 2020, 56(11): 118-123.

[12] 张平, 栗亚敏. 前向安全的椭圆曲线数字签名方案[J]. 计算机工程与应用, 2020, 56(1): 115-120.

[13] Chaum, D. and Van Heyst, E. (1991) Group Signatures. Advances in Cryptology-EUROCRYPT'91. Springer-Verlag, Berlin, 257-265. https://doi.org/10.1007/3-540-46416-6_22

[14] 苏吟雪, 田海博. 基于 SM2 的双方共同签名协议及其应用[J]. 计算机学报, 2020, 43(4): 701-710.