计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(4):296−301 [doi: 10.15888/j.cnki.csa.009473]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 基于容忍泄露的内容关联短签名方案①

左黎明, 周　婷, 刘晨宁

(华东交通大学 理学院, 南昌 330013)
通信作者: 左黎明, E-mail: limingzuo@126.com

**摘　要**: 容忍泄露是指为了增强签名方案的鲁棒性, 允许方案泄露部分秘密信息, 适用于设备和通讯线路均没法完美保护的大部分场合. 短签名长度一般只有普通签名一半的签名长度, 可以大大降低窄带实时交互的系统的通讯数据量. 提出了一种待签名信息关联签名密钥的短签名方案, 该方案具有容忍部分泄露的特性, 对方案的效率和安全性进行了分析, 证明了方案在容忍泄露预言机下是安全的, 实验结果表明该方案具有较好的性能, 适用于传输带宽受限的应用场合.

**关键词**: 容忍泄露; 短签名; 内容关联密钥; 可证安全

引用格式: 左黎明,周婷,刘晨宁.基于容忍泄露的内容关联短签名方案.计算机系统应用,2024,33(4):296–301. http://www.c-s-a.org.cn/1003-3254/9473.html

## Leakage Tolerance-based Content-associated Short Signature Scheme

ZUO Li-Ming, ZHOU Ting, LIU Chen-Ning

(School of Science, East China Jiaotong University, Nanchang 330013, China)

**Abstract**: Leakage tolerance refers to allowing the scheme to leak some secret information to enhance the robustness of the signature scheme, which is suitable for most occasions where the equipment and communication lines cannot be perfectly protected. The length of the short signature is generally only half that of the ordinary signature, which can greatly reduce the communication data volume of the narrowband real-time interactive system. This study proposes a short signature scheme for the signature key associated with the information to be signed, and the scheme is tolerant to partial leakage. The efficiency and security of the scheme are analyzed, and the security of the scheme is proved under the tolerant leak oracle. The experimental results show that the scheme has good performance and is suitable for applications with limited transmission bandwidth.

**Key words**: leakage tolerance; short signature; content-associated key; provable security

传统数字签名中通常假设攻击者无法获取秘密信息, 但在实际应用中并非如此. 边信道攻击[1]可以利用设备运行时的电磁辐射特征, 分析设备的边信道信息如功耗、电磁辐射等推断出秘密参数和密钥. 故障注入攻击[2]可以向某些嵌入式系统注入故障信号获取系统写保护的敏感参数信息. 冷启动攻击[3]在系统或应用程序初始化和操作等过程中仍能获取系统内部的部分秘密信息. 真实数据传输过程[4–6]中通常也没有真正意义上的完美安全的信道用来传递秘密参数或者密钥, 特别是野外硬件设施和通讯网络无法做到真正意义上的物理隔离, 比如铁路和公路的实时传感控制系统网络[7,8], 因此容忍泄露的数字签名方案在实际中具有重要的应用价值.

短签名方案由 Boneh 等人[9]首次提出, 方案签名

① 基金项目: 江西省教育厅科技项目 (GJJ200626, GJJ210625)
收稿时间: 2023-09-01; 修改时间: 2023-10-25, 2023-11-09; 采用时间: 2023-12-15; csa 在线出版时间: 2024-03-04
CNKI 网络首发时间: 2024-03-08

长度是 DSA (digital signature algorithm) 签名的一半. 文献[10]基于 Boneh 等人[9]短签名和 Gap-Diffie-Hellman 群提出了新型短签名方案, 缩减了签名运算次数, 并在随机预言模型下证明了方案的适应性选择密文攻击的安全性, 文献[11]构造一种可证安全短盲签名方案, 并证明了方案安全性, 方案适用于带宽受限的数据传输场景, 文献[12]提出了一种基于双线性映射的短代理签名, 并基于 Diffie-Hellman 问题和选择密文攻击证明了方案的安全性. 文献[13]结合公钥密码体制提出了一种基于证书的短签名方案, 并证明了方案的安全性和运行效率. 文献[14]针对代理签名重签名方案存在的效率缺陷, 结合短盲签名思想提出了短盲代理重签名方案的定义. 文献[15,16]基于短签名构造了高效的数据完整性签名验证方案. 文献[17–19]的方案通过结合短签名减少签名运算量, 提升了方案计算效率.

这些方案没有考虑实际中可能发生的敏感信息[20] (例如部分秘钥和秘密参数) 可以泄露的情形, 在以前工作的基础上, 本文提出了一种安全的容忍泄露的内容关联短签名方案, 可以容忍部分秘密信息泄露并做到签名与待签名内容强关联, 提高签名方案的安全性.

## 1 相关基础与安全模型

### 1.1 数学基础

定义 1. 双线性对映射. $G_1$和$G_2$是椭圆曲线上$q$阶循环加法群, $P$是$G_1$的生成元, $Q$是$G_2$的生成元, $G$是乘法循环群, 则有双线性映射$e=G_1\times G_2\to G$满足以下性质.

(1) 双线性性: 对于任意的$a,b\in Z_q^*$, 有$e(aP,bQ)=e(P,Q)^{ab}$.

(2) 非退化性: $e(P,Q)\neq 1$.

(3) 可计算性: 存在多项式算法计算$e(P,Q)$的值.

定义 2. 计算 Diffie-Hellman 问题 (CDH). 椭圆曲线群$G$, 其阶为$q$, $P$为生成元, 给定$a,b\in Z_q^*$, $aP$, $bP$, 求$abP$.

### 1.2 容忍泄露

定义 3. 容忍泄露限. 在一个签名或者加密方案中, 秘密参数的总信息量是与安全参数$k$正相关的一个常量$I_0$, 如果存在一个阈值$\lambda<I_0$, 在泄露信息量小于$\lambda$的情况下方案仍然是安全的, 则称该方案为$\lambda$-容忍泄露方案. 称$\lambda$为容忍泄露限.

传统的构造容忍泄露的方案的思路是利用秘密分割方案把秘密参数分成很多碎片, 只要没有收集到足够多的碎片, 无法恢复出秘密信息和参数, 方案就是安全的. 本文我们采用内容关联多组秘钥的方式实现容忍泄露, 虽然增加了存储成本, 但计算量远小于传统基于门限分割的签名方案, 且具有较大的容忍泄露限. 只要$n$组密钥中有 1 组密钥没有泄露, 泄露信息量小于$2(n-1)k$, $\lambda=2(n-1)k$则方案是安全的, 其中, $k$为单个私钥长度.

### 1.3 敌手模型

容忍泄露模型假设攻击者可获得系统部分秘密信息, 如部分密钥等. 因此敌手模型中需要提供泄露询问.

敌手$A$与挑战者$C$之间的游戏如图 1 所示. 挑战者$C$要借用敌手$A$求解困难问题.

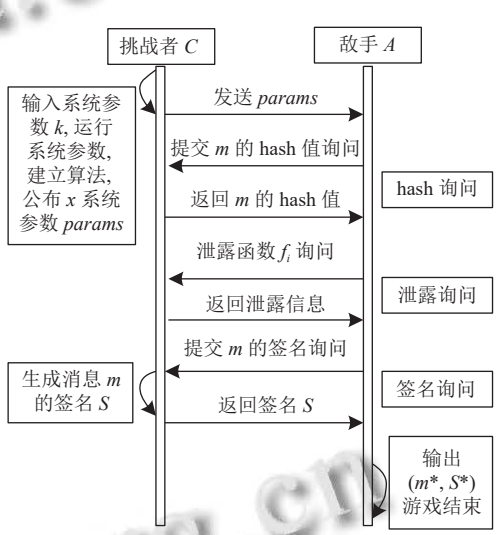


图 1 抗泄露模型交互过程

(1) 系统建立: 挑战者$C$运行系统生成算法, 生成系统的公开参数、秘密参数、私钥和公钥, 公布系统参数$params$发送给敌手$A$.

(2) 询问阶段: 敌手$A$对$C$进行多次选择适应性询问.

1) hash 询问: $A$询问消息$m$的哈希值, $C$返回$H(m)$.

2) 泄露询问: $A$询问$m$的泄露信息, $C$返回泄露的部分信息.

3) 签名询问: $A$询问消息$m$的签名, $C$生成的签名$S$返回给敌手$A$.

(3) 伪造阶段: $A$结束询问并输出伪造的签名$(m^*,S^*)$, 游戏结束.

## 2 签名方案与安全性分析

### 2.1 签名方案

传统签名方案使用相同私钥对不同消息摘要进行

签名, 很容易在电磁辐射频谱中出现稳定的辐射特征. 本文通过一个哈希函数实现待签名消息与签名密钥相关联, 这样只要待签名消息改变就会造成使用的签名密钥不一样, 极大地增强了签名方案的容忍泄露能力. 方案构造主要如下.

(1) 系统参数建立. 选取安全参数$k$, 秘钥组参数$n$, 选择阶为素数$q$的椭圆曲线上的循环加法群$G_1$($P$为$G_1$的生成元), 循环乘法群$G_2$, 非退化的双线性映射$e=G_1\times G_1\to G_2$, 选择安全抗碰撞哈希函数$H_1:\{0,1\}^*\to\{0,1\}^{len}$, $H_2:\{0,1\}^*\to G_1$, 其中哈希函数$H_1(m)$的哈希长度$len>n$, $H_1(m)=b_1b_2\cdots b_{len}$, $b_i\in\{0,1\}$, 公开系统参数$params=\{k,G_1,G_2,P,H_1,H_2\}$.

(2) 密钥生成. 随机生成签名者的私钥组$x=\begin{Bmatrix}x_{10},x_{20},\cdots,x_{n0}\\x_{11},x_{21},\cdots,x_{n1}\end{Bmatrix}$, 计算签名者公钥组$y=xP=\begin{Bmatrix}y_{10},y_{20},\cdots,y_{n0}\\y_{11},y_{21},\cdots,y_{n1}\end{Bmatrix}$, 其中$x_{ij}\in Z_q^*$, $y_{ij}=x_{ij}P$, 公开公钥组.

(3) 签名算法. 签名者对消息$m\in\{0,1\}^*$进行签名, 签名过程如下.

1) 计算$x_m=\sum_{i=1}^{n}x_{ib_i}$, 其中$x_{ib_i}$表示当$b_i=1$时, 取$x_{i1}$, 当$b_i=0$时, 取$x_{i0}$.

2) 计算$S=x_mH_2(m)$, $S$为签名者对消息$m$的签名, 将$S$发送给签名验证者.

(4) 验证算法. 对给定的消息和对应签名$S$进行验证, 过程如下.

1) 计算$H_1(m)=b_1b_2\cdots b_{len}$, $y_m=\sum_{i=1}^{n}y_{ib_i}$.

2) 验证$e(S,P)=e(H_2(m),y_m)$.

正确性有以下恒等式保证, 当且仅当等式验证成立接受签名.

$$\begin{aligned}e(S,P)&=e(x_mH_2(m),P)\\&=e(H_2(m),x_mP)\\&=e\left(H_2(m),\sum_{i=1}^{n}x_{ib_i}P\right)\\&=e\left(H_2(m),\sum_{i=1}^{n}y_{ib_i}\right)\\&=e(H_2(m),y_m)\end{aligned}$$

### 2.2 签名方案安全性分析

定理 1. 本文方案在 CDH 困难问题假设和容忍 ROM 模型下是可证安全的.

引理 1. 若算法$A$在概率多项式时间$t$内, 以一个不可忽略的优势$\varepsilon$, 成功伪造签名, 则存在概率多项式时间算法$C$, 在时间$t'<t+(q_st_s+q_dt_d+2q_{H_1}t_{H_1}+2q_{H_2}t_{H_2})$内以不可忽略的优势解决 CDH 问题:

$$\varepsilon'\geqslant\left(\varepsilon-\frac{1}{2^k}\right)\cdot\left(1-\frac{1}{q_d}\right)\cdot\left(1-\frac{1}{q_S}\right)\cdot\frac{1}{q_{H_1}}\cdot\frac{1}{q_{H_2}}$$

其中, 参数意义如下: $q_d$: 泄露询问次数 ($q_d<n$), $t_d$: 询问一次所需时间, $q_{H_1}$: 询问$H_1$预言机次数, $t_{H_1}$: 一次询问所需时间, $q_{H_2}$: 询问$H_2$预言机次数, $t_{H_2}$: 一次询问所需时间, $q_s$: 签名询问次数, $t_s$: 一次询问所需时间.

证明: 给定一个 CDH 困难问题的实例: $G_1$是椭圆曲线上$q$阶循环加法群, $P$是$G_1$的生成元, 输入$aP\in G_1$, $bP\in G_1$, 输出$abP$.

算法$C$需要通过调用算法$A$来求解 CDH 问题. 不妨设$A$不会做两次相同的询问, 并在每轮游戏一开始初始化清空所有记录列表. $C$选取安全参数$k$, 秘钥组参数$n$, 随机生成$2n-2$个秘密参数$x_{ij}\in Z_q^*$ ($1\leqslant i\leqslant n-1$)构建签名者的私钥组$x=\begin{Bmatrix}x_{10},x_{20},\cdots,x_{n0}\\x_{11},x_{21},\cdots,x_{n1}\end{Bmatrix}$, 这里需要说明的是$i=n$时$x_{n0}$, $x_{n1}$位置本来应该是$a$, 但因为$a$未知, 因此填入“$\perp$”表示空缺. 对应的公钥组为$y=\begin{Bmatrix}y_{10},y_{20},\cdots,aP\\y_{11},y_{21},\cdots,aP\end{Bmatrix}$, 即当$1\leqslant i\leqslant n-1$时$x_{ij}\in Z_q^*$, $y_{ij}=x_{ij}P$, 当$i=n$时, $y_{n0}=y_{n1}=aP$, 将实例的输入条件之一$aP_1$嵌入在指标为$i$的公钥之中, $C$秘密保存$x$但允许$A$做$q_d$次不包括$x_{n0}$和$x_{n1}$的私钥泄露询问. $C$在系统初始化后将系统参数$params=\{k,n,G_1,G_2,q,P,y,H_1,H_2\}$发送给$A$, 交互中会选择一个消息$m^*$为目标消息在其哈希值中嵌入$bP$.

(1) $H_1$询问: $C$维护表$LH_1$, $LH_1$由数组$(m,w)$组成, $A$向$C$提交关于消息$m$的$H_1$询问时, 如果$(m,w)$已经在$LH_1$中, 直接返回$w$给$A$, 否则随机选择一个$w=b_1b_2\cdots b_{len}$返回给$A$同时将$(m,w)$记录到$LH_1$.

(2) $H_2$询问: $C$维护一个列表$LH_2$, $LH_2$由数组$(m,u,h)$组成, 当$A$向$C$提交一个消息$m$的$H_2$询问时, 如果$(m,u,h)$已经在$LH_2$中, 直接返回$h$给$A$, 否则执行以下操作.

1) 当$m\neq m^*$时随机选择一个$u\in Z_q^*$, $h=uP$, 将$h=uP$返回给$A$同时将$(m,u,h)$记录到$LH_2$.

2) 当$m=m^*$时, 令$h=bP$, 将$h$返回给$A$同时将$(m,\perp,h)$记录到$LH_2$.

(3) 泄露询问: $C$维护一个由数组$(i,j,x_{ij})$组成的列

表$L_{sk}$. 当$A$向$C$提出关于指标$(i,j)$泄露部分私钥询问时, $C$执行如下操作.

1) 如果表$L_{sk}$中已有指标$(i,j)$相应记录$(i,j,x_{ij})$, 则返回$x_{ij}$给$A$.

2) 如果$i\neq n$, $C$将$x_{ij}$返回给$A$的同时将$(i,j,x_{ij})$保存到表$L_{sk}$中.

3) 如果$i=n$, $C$返回“$\perp$”给$A$同时将$(i,j,\perp)$保存到表$L_{sk}$中, 此事件记录为$E_1$.

(4) 签名询问: $A$向$C$输入待签名消息$m$, $C$从$LH_2$中恢复$(m,u,h)$, 然后进行以下操作.

1) 如果$m\neq m^*$, 则$C$从列表$LH_1$获得对应数组$(m,w)$, 此时$w=b_1b_2\cdots b_{len}$, $b_i\in\{0,1\}$, $h=uP$计算$x_m=\sum_{i=1}^{n-1}x_{ib_i}$, $S=x_muP+uaP$, 则$S$即为身份为$ID$对消息$m_{ID}$的签名. $C$将$S$返回给$A$.

2) 如果$m=m^*$, $C$终止模拟, 输出“$\perp$”(记该事件为$E_2$).

经过多项式有界次询问后, $A$输出一个关于消息$m^*$且满足最终验证等式使得$Verify(m^*,S^*)=1$的消息/签名对$(m^*,S^*)$, 用$E_3$表示该事件. $C$从列表$LH_1$中恢复$(m^*,w^*)$和列表$LH_2$中恢复数组$(m^*,u^*,h^*)=(m^*,\perp,bP)$, 因为签名验证等式$e(S^*,P)=e(H_2(m^*),y_{m^*})$成立, 则有以下等式成立:

$$e(S^*,P)=e(x_{m^*}H_2(m),P)=e(H_2(m^*),x_{m^*}P)$$
$$=e\left(bP,aP+\sum_{i=1}^{n-1}x_{ib_i}P\right)=e\left(P,abP+\sum_{i=1}^{n-1}x_{ib_i}bP\right)$$

$C$计算$S^*-\sum_{i=1}^{n-1}x_{ib_i}bP=abP$, 输出$S^*-\sum_{i=1}^{n-1}x_{ib_i}bP$作为实例的解答. $C$解决 CDH 问题的时间和优势.

(1) $A$对消息都经过了$H_1$询问和$H_2$询问, 并且答案是有效的.

(2) $E_1$, $E_2$不发生时, 游戏才能正常完成.

(3) $E_1$, $E_2$都不发生时, $E_3$发生, 则$C$能解决 CDH 问题的一个实例. 则$E_1$和$E_2$都不发生的概率为:

$$\Pr(\neg E_1\wedge\neg E_2)=\left(1-\frac{1}{q_d}\right)\left(1-\frac{1}{q_S}\right)$$

当$A$没询问$H_2$的概率为$\frac{1}{2^k}$, 所以$C$的优势下界为:

$$\varepsilon'\geqslant\left(\varepsilon-\frac{1}{2^k}\right)\cdot\left(1-\frac{1}{q_d}\right)\cdot\left(1-\frac{1}{q_S}\right)\cdot\frac{1}{q_{H_1}}\cdot\frac{1}{q_{H_2}}$$

运行时间的一个上界为:

$$t'<t+(q_st_s+q_dt_d+2q_{H_1}t_{H_1}+2q_{H_2}t_{H_2})$$

其中, $t'$为概率多项式时间, $\varepsilon'$为不可忽略的概率. 因此$C$在$t'$内以$\varepsilon'$成功地求解了 CDH 问题实例.

由引理 1 即证定理 2, 说明本文方案可以抵抗适应性选择消息攻击下的存在性伪造攻击.

本文提出的短签名方案由于构造的抗泄露短签名方案的签名密钥是内容关联密钥, 通过提取签名消息的部分敏感信息生成与原始文件内容相关的密钥, 密钥与特定内容相关且具有单向性, 只能用于特定的加密和解密内容, 抵御了密钥破解的相关推断攻击.

另外方案将私钥表示为密钥组形式, 恶意中间人若要成功伪造消息签名, 必须达到泄露门限值, 否则无法完整恢复私钥进行签名, 由方案可知获取密钥组元素是困难的, 所以当部分私钥泄露时, 整体安全性不会受到影响, 签名私钥具有抗泄露性. 另外方案签名时对数据进行了哈希计算, 若传输检测数据受到篡改, 则会引起哈希值的变化, 导致签名验证失败, 因此方案可抵抗消息篡改攻击.

## 3 实现与效率比较

### 3.1 方案代码实现

本文在 64 位 Windows 10 操作系统的 IntellijIDEA 2020 开发平台下, 为了方便比较, 使用了国密 SM2 和 SM9 算法相同的椭圆曲线和双线性对, 使用 SM3 做哈希运算, 使用 Java 语言 (JDK 版本为 1.8) 实现了本文短签名方案, 图 2 为方案实验结果, 方案主要代码如算法 1.

算法 1. 本文短签名方案主要代码

```
//循环计算 50 次
for (int i =0; i<50; i++){
  //(1) 初始化参数
  sm9Curve sm9Curve = new sm9Curve();
  sm9 sm9 = new sm9(sm9Curve);
  sm3 sm3 = new sm3();
  Pairing bp = sm9.getCurve().sm9Pairing;
  byte[] hash1;
  int n = 10;
  Element g = sm9.getCurve().P2; //生成元 g
  Map<String, Element[][]> keyMap = getKeys(bp, n, g);
  Element[][] priKeys = keyMap.get("pri"); //生成私钥组
  Element[][] pulKeys = keyMap.get("pul"); //生成公钥组
  stimeh = System.currentTimeMillis(); //计算执行时间
  //(2) 签名运算
  hash1 = sm3.hash(m);
  boolean[] h1 = byteToBoolean(hash1);
```

```
h2 = bp.getG1().newElementFromHash(hash1, 0, hash1.length);
Map<String, Element> xyMap = getXY(bp, priKeys, pulKeys, n, h1, g);
Element xm = xyMap.get("xm");
Element ym = xyMap.get("ym");
Element S = h2.mulZn(xm); //签名生成
stimeh1 = System.currentTimeMillis(); //计算执行时间
signtime += stimeh1 – stimeh; //计算签名时间
//(3) 签名验证
left = bp.pairing(S, g); //计算等式左边
right = bp.pairing(h2, ym); //计算等式右边
stimeh2 = System.currentTimeMillis();
  verifytime += stimeh2 – stimeh1;//计算验证时间
}
long etime = System.currentTimeMillis(); // 计算执行时间
System.out.println("待签名消息 m:" + m);
System.out.println("等式左边 e(S, P)=\n" + left);
System.out.println("等式右边 e(H2(m), ym)=\n" + right);
if (left.isEqual(right)) {
  System.out.println("签名验证成功!");
} else {
  System.out.println("验证失败!");
}
```

```
待签名消息 m:happybirthday
等式左边 e(S,P)=
[{x=749871045498097754947767217749401123041405145463677057896256083921677068380 51,y=790417569506291423691903112118665618770138050072298991053396244603798681 54887},
 {x=136001701959090373749953452299444233689076167866111716956462049587680887796 082,y=155230029197939045285364933917714338016611970249175051603757239531743336 14534},
 {x=957285795288142603556433381294176983184779152515608892855692412256548610438 82,y=709202153263046759554515661933611867598966872647211314272501437308103941 86443},
 {x=258741727176295656202352519208822164333420953826813362527368727477333757506 90,y=509677490282052904748687003541557916668392138244936799262246621031637833 31274},
 {x=179320819817777976870031106717093396504898290824611922596602654429637507793 339,y=592307946397782833920686205668130012564028464316193999498900389767779316 303758},
 {x=255278387540230240829670560184563270697988425344291038684356411213479811248 35,y=533885131516180299249737903761869137082687492982926951171271003380281239 39020}, ]
等式右边 e(H2(m),ym)=
[{x=749871045498097754947767217749401123041405145463677057896256083921677068380 51,y=790417569506291423691903112118665618770138050072298991053396244603798681 54887},
 {x=136001701959090373749953452299444233689076167866111716956462049587680887796 082,y=155230029197939045285364933917714338016611970249175051603757239531743336 14534},
 {x=957285795288142603556433381294176983184779152515608892855692412256548610438 82,y=709202153263046759554515661933611867598966872647211314272501437308103941 86443},
 {x=258741727176295656202352519208822164333420953826813362527368727477333757506 90,y=509677490282052904748687003541557916668392138244936799262246621031637833 31274},
 {x=179320819817777976870031106717093396504898290824611922596602654429637507793 339,y=592307946397782833920686205668130012564028464316193999498900389767779316 303758},
 {x=255278387540230240829670560184563270697988425344291038684356411213479811248 35,y=533885131516180299249737903761869137082687492982926951171271003380281239 39020}, ]
签名验证成功!
签名方案签名时长：6 毫秒
签名方案签名验证时长：643 毫秒
签名方案执行总时长：917 毫秒
```

图 2 方案运行结果

### 3.2 方案性能分析

本方案选取了经典短签名方案和近年来提出的短签名方案. 这些方案基于盲签名、代理签名以及多重短签名实现, 具有签名效率高、计算量小等特点. 这里从签名长度、计算复杂度以及安全性等方面与本文方案进行性能上的分析, 表 1 记录了各方案的签名计算效率和签名长度. $H$ 为hash计算, $G_1$ 为加法群, $M$ 为 $G_1$ 上的倍点标量乘, $P_r$ 为双线性对运算, $E$ 为幂乘运算, $|G_1|$表示$G_1$上元素的长度, $|Z_q^*|$表示$Z_q^*$上元素的长度.

表 1 各方案性能对比

| Schemes | Signature phase | Verification phase | Length |
|---|---|---|---|
| 文献[9] | $1M+1H$ | $2P_r+1H$ | $\|G_1\|$ |
| 文献[11] | $1M$ | $2P_r+1H$ | $\|G_1\|$ |
| 文献[12] | $1M+1H$ | $2P_r+1M+2H$ | $\|G_1\|$ |
| 文献[21] | $1P_r+1E$ | $1P_r+2M$ | $\|G_1\|+2\|Z_q^*\|$ |
| 文献[22] | $2M+1H$ | $3P_r+2M+2H$ | $2\|G_1\|$ |
| 文献[23] | $2M$ | $2P_r+1M$ | $\|G_1\|$ |
| 本文 | $1M+1H$ | $2P_r+1H$ | $\|G_1\|$ |

由表 1 可知, 本方案在签名阶段只需要 1 次标量乘 $M$ 和 1 次hash计算 $H$, 验证阶段需 2 次双线性对运算 $P_r$ 和 1 次hash运算 $H$, 签名长度为 $|G_1|$. 根据对比结果分析, 本方案使用的签名长度短于文献[21,22], 与其他短签名方案[9,11,12,23]长度保持一致, 签名阶段计算复杂度低于文献[12,21–23], 验证阶段计算复杂度低于文献[12,22], 本方案理论上整体计算效率与性能与文献[9,11]接近一致. 虽然椭圆曲线对运算相对倍乘运算效率略低, 但该方案优势主要在签名长度上, 且椭圆曲线运算交互次数少, 为了保证抗泄露性牺牲了一些存储空间和性能, 所以综上本方案适用于窄带传输场景.

### 3.3 运行效率分析

在相同实验环境下实现以上几种方案, 计算运行 50 次完成签名所耗时间, 得到的效率比较如表 2 所示. 由表中结果可知, 本方案签名过程平均耗时 0.006 s, 验证过程平均耗时 0.643 s, 方案总平均耗时 0.917 s, 方案总耗时与文献[9,11]耗时接近一致, 比文献[12,21–23]分别减少了 35%、40%、48% 和 43%. 总体来看, 本方案整体计算耗时低, 运行效率更高.

表 2 各方案运行时间对比 (s)

| Schemes | Signature average | Veritification average | Total average |
|---|---|---|---|
| 文献[9] | 0.006 | 0.640 | 0.734 |
| 文献[11] | 0.009 | 0.651 | 0.794 |
| 文献[12] | 0.005 | 0.617 | 1.403 |
| 文献[21] | 0.010 | 0.661 | 1.517 |
| 文献[22] | 0.035 | 0.939 | 1.755 |
| 文献[23] | 0.056 | 1.403 | 1.601 |
| 本文 | 0.006 | 0.643 | 0.917 |

## 4 总结

本文提出了一种基于容忍泄露的内容关联密钥短签名方案, 并在随机预言模型下证明了方案的安全性. 将本文方案与其他方案进行实验对比, 由实验比较结果可以得出结论, 本文方案具有较短的签名长度, 计算效率高, 并能抵抗泄露攻击, 适用于各种需要数据安全快速传输的计算场景. 在下一步研究中, 将利用盲签名的构造思想, 设计一套适用于数据安全传输系统的容忍泄露盲签名方案.

## 参考文献


1 Wu JD, Tseng YM, Huang SS, *et al*. Leakage-resilient certificate-based signature resistant to side-channel attacks. IEEE Access, 2019, 7: 19041–19053. [doi: 10.1109/ACCESS.2019.2896773]

2 周彦伟, 马岿, 乔子芮, 等. 基于证书的抗连续泄露签名机制. 计算机学报, 2022, 45(11): 2363–2376.

3 Zhou YW, Yang B, Wang T, *et al*. Continuous leakage-resilient certificate-based encryption scheme without bilinear pairings. The Computer Journal, 2020, 63(4): 508–524. [doi: 10.1093/comjnl/bxz085]

4 Nagravision SA. Securing digital data transmission in a communication network: US16789078[P/OL]. 2022-06-28.

5 Zhang HT, Zhang LC, Guo Y, *et al*. Data transmission mechanism of vehicle networking based on fuzzy comprehensive evaluation. Open Mathematics, 2022, 20(1): 1909–1925. [doi: 10.1515/math-2022-0537]

6 Munir N, Khan M, Hussain I, *et al*. Cryptanalysis of encryption scheme based on compound coupled logistic map and anti-codifying technique for secure data transmission. Optik, 2022, 267: 169628. [doi: 10.1016/j.ijleo.2022.169628]

7 金鹏, 黄浩, 刘检华, 等. 多传感器信息融合的铁路扣件缺陷检测方法. 机械工程学报, 2021, 57(20): 38–46.

8 Wang SH, Wang TL, Pei X, *et al*. Highway icing time prediction with deep learning approaches based on data from road sensors. Science China Technological Sciences, 2023, 66(7): 1987–1999. [doi: 10.1007/s11431-022-2230-8]

9 Boneh D, Lynn B, Shacham H. Short Signatures from the Weil Pairing. Journal of Cryptology, 2004, 17(4): 297–319. [doi: 10.1007/s00145-004-0314-9]

10 杜红珍, 温巧燕. 高效的短签密方案. 北京邮电大学学报, 2008, 31(1): 84–87. [doi: 10.3969/j.issn.1007-5321.2008.01.020]

11 左黎明, 夏萍萍, 陈祚松. 一种可证安全的短盲签名方案. 计算机工程, 2019, 45(12): 114–118.

12 左黎明, 陈祚松, 夏萍萍, 等. 高效的可证安全短代理签名方案. 计算机应用, 2018, 38(12): 3455–3461.

13 左黎明, 陈兰兰, 周庆. 一种基于证书的短签名方案. 山东大学学报 (理学版), 2019, 54(1): 79–87.

14 Yang XD, Ma TC, Chen CL, *et al*. An efficient short blind proxy re-signatures scheme. Journal of Physics: Conference Series, 2019, 1302(2): 022006. [doi: 10.1088/1742-6596/1302/2/022006]

15 Zhu HL, Yuan Y, Chen YL, *et al*. A secure and efficient data integrity verification scheme for cloud-IoT based on short signature. IEEE Access, 2019, 7: 90036–90044. [doi: 10.1109/ACCESS.2019.2924486]

16 常亮, 王冠棋, 杨雪欣. 一种云存储完整性的格签名验证方法. 黑龙江科技大学学报, 2020, 30(4): 455–459.

17 张君何, 周清雷, 韩英杰. 一种基于环签名和短签名的可净化签名方案. 计算机科学, 2020, 47(S1): 386–390, 399.

18 Zou LN, Wang XY, Deng LF. Secure data fusion analysis on certificateless short signature scheme based on integrated neural networks and elliptic curve cryptography. EAI Endorsed Transactions on Scalable Information Systems, 2022, 9(34): e3.

19 Yang ZY, Wang ZQ, Qiu F, *et al*. A group key agreement protocol based on ECDH and short signature. Journal of Information Security and Applications, 2023, 72: 103388. [doi: 10.1016/j.jisa.2022.103388]

20 Yu QH, Li JG, Ji S. Identity-based and leakage-resilient broadcast encryption scheme for cloud storage service. Applied Sciences, 2022, 12(22): 11495. [doi: 10.3390/app122211495]

21 Islam SKH, Biswas GP. A provably secure identity-based strong designated verifier proxy signature scheme from bilinear pairings. Journal of King Saud University-Computer and Information Sciences, 2014, 26(1): 55–67. [doi: 10.1016/j.jksuci.2013.03.004]

22 Hu XM, Tan W, Xu HJ, *et al*. Short and provably secure designated verifier proxy signature scheme. IET Information Security, 2016, 10(2): 69–79. [doi: 10.1049/iet-ifs.2014.0434]

23 左黎明, 陈兰兰, 周庆. 一种适用于分布式审批工作流的多重短签名方案. 计算机应用研究, 2020, 37(2): 521–525.


(校对责编: 孙君艳)