基于正交拉丁方理论的数字签名分组批量验证

王宏[1]，赖成喆[2]，刘向阳[1]，曾晗[1]
（1. 国防科技大学信息通信学院，陕西 西安 710106；2. 西安邮电大学网络空间安全学院，陕西 西安 710121）

摘 要：针对态势感知网络中海量的、时敏性强的消息需要中心节点进行快速、安全验证的问题，基于组合数学的正交拉丁方理论设计了数字签名分组批量认证方案。该方案着眼于消息安全性验证效率的提升，利用正交拉丁方理论设计数字签名的分组方案，以聚合签名理论作为签名批量验证算法，构建了一个采取多个处理器并行运算的非适应性数字签名分组批量验证模型。理论证明和仿真分析表明，所提模型在非法数字签名个数上限 d（$d \ll n$）已知的条件下，能以大约 $\sqrt{n}$ 次数完成 n 个非法数字签名的识别，特别是在多个处理器计算的情况下相比逐一验证、二分法验证具有时效高、容错性强的特点。

关键词：数字签名；正交拉丁方；分组设计；批量验证
中图分类号：TN918
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022036

Orthogonal Latin square theory based group and batch verification for digital signatures

WANG Hong[1], LAI Chengzhe[2], LIU Xiangyang[1], ZENG Han[1]
1. College of Information and Communication, National University of Defense Technology, Xi'an 710106, China
2. School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Abstract: In order to solve the problem of fast security verification of massive and time-intensive messages on a central node in situational awareness networks, orthogonal Latin square theory based scheme was considered. Considering efficiency promotion of security verification of messages, group design of digital signatures based on orthogonal Latin square theory was formulated, batch verification of digital signatures was processed by aggregate signature, then an efficient, parallel and non-adaptive batch verification scheme of digital signatures was proposed in according with multiple processors. Theoretical analysis and simulation results demonstrate that it will be able to identify n digital signatures by approximately $\sqrt{n}$ times given the upper bound d ($d \ll n$) of invalid digital signatures, together with higher time-efficiency and stronger error-tolerance by comparing with individual testing and binary splitting algorithms especially when multiple processors are available.

Keywords: digital signatures, orthogonal Latin square, group design, batch verification

0 引言

随着无线通信、传感器及人工智能等技术的不断进步，各种态势感知网络在气象水文监测、城市智慧交通、能源在线监测等方面发挥着越来越重要的作用。技术进步带来便捷的同时，也导致人们受制于技术，尤其是个人信息的被盗、冒充或伪造等现象时有发生，极易造成态势感知网络的不畅，甚至瘫痪。因此，如图 1 所示，对网络传递的消息进行签名，确保消息的可靠性、完整性和不可抵赖性成为一种重要的网络安全措施。传统的数字签名采用逐一验证方法，当中心型节点验证的签名数量较

收稿日期：2021−12−01；**修回日期**：2022−01−18
基金项目：国家自然科学基金资助项目（No.61871471）；陕西省重点研发计划基金资助项目（No.2021ZDLGY06-02）
Foundation Items: The National Natural Science Foundation of China (No.61871471), The Key Research and Development Program of Shaanxi Province (No.2021ZDLGY06-02)

少时，逐一验证方法尚且可以适应要求；当需要验证的签名数量巨大时，逐一验证需要消耗大量时间，导致大量消息由于不能及时得到验证而被迫丢弃，比如庞大的车联网系统每隔 100~300 ms 就要实时地进行消息传输，大量的消息连同附带的签名涌向中心节点等待马上验证[1-2]，以保证它们来源的可靠性。因此，有关数字签名的快速验证算法的研究成为近年密码学的研究热点。

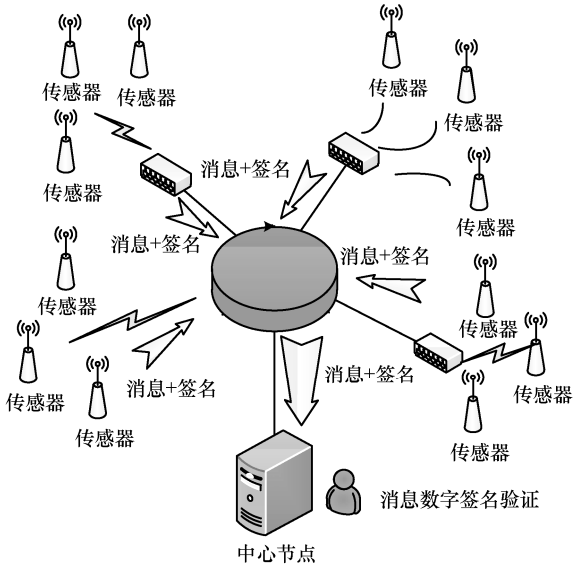


图 1　中心型网络消息汇聚

围绕数字签名快速验证算法，研究者所做的工作归纳为 2 个方面：一是数字签名的批量验证算法研究；二是分组检测方案设计。数字签名的批量验证通常采取聚合签名算法，聚合签名是将具有同态结构的签名算法进行聚合验证，达到一次性验证多个数字签名的目的，基于模指数运算的数字签名算法（DSA, digital signature algorithm）就具有这种同态结构[3-7]。近年来，许多聚合签名研究成果实现并证明了签名密钥的生成、分发和数字签名的聚合验证[8-11]，但聚合签名只能批量检测多个数字签名的集合中是否存在非法数字签名，不能鉴别非法数字签名在集合中的具体位置。为此，研究者纷纷引入已在生物医学检测中广泛应用的分组检测理论进行数字签名的分组检测。当数字签名集合中存在非法数字签名时，通过将数字签名分成不同的组，分组实施批量验证，从而逐步确定非法数字签名。

分组检测方案是针对数字签名聚合验证存在至少一个非法数字签名而进行非法对象识别的分组聚合验证方法，按照分组之间的关系，可以分为序贯类分组检测和非适应性分组检测[12]。序贯类分组检测是按照一定规则对检测对象进行“多次”有序分组检测，直到所有非法对象被全部识别。但在序贯类分组检测中，前一次检测的结果决定后一次分组的情况，检测过程具有严格的顺序性，只能一步一步按照分组顺序完成，如基于二元分叉树的分组检测。非适应性分组检测是根据一定规则巧妙地将检测对象“一次性”分成若干组，同一检测对象可以包含在多个组中，使检测具有一定数量的对象析取性，便于采用平行检测的方法对各组进行同时检测，并通过检测结果的呈现情况析取非法数字签名[3]。序贯类分组检测简单、实现容易，但运算效率很难提高[13-14]。非适应性分组检测可以并行检测，在多处理器的情况下，运算效率较高，但分组方案构造复杂。现有的非适应性分组检测算法具有以下 3 个特点：一是分组检测的次数、时间等参数界限研究探索较多[15-16]，而具体方案构造研究成果较少；二是分组检测方案构造不准确，存在非法数字签名不能完全析出的情况，如随机矩阵法[12]；三是分组检测方案的存在性理论缺乏严格证明，如基于纠错码（ECC, error correction code）的分组方案构建，缺少检测对象数量变化时的分组方案存在性证明[3,17-18]。

综上所述，本文针对数字签名的快速验证问题，研究分组批量验证算法，基于拉丁方理论构造分组方案，以较少次数的群组认证完成非法数字签名的识别，实现多个消息数字签名的快速验证。本文主要贡献如下。

1) 采用拉丁方理论构建横截设计。以有限域理论为基础设计素数幂阶拉丁方，证明素数幂阶拉丁方存在性是横截设计存在的充分必要条件，为数字签名分组批量验证方案确立提供理论基础。

2) 基于横截设计理论构建析取矩阵。证明当横截设计 TD$[k,\lambda;m]$的 $\lambda = 1$ 时，其关联矩阵 $\boldsymbol{M}$ 的转置矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 是一个析取矩阵，将区组设计与析取矩阵联系起来，为析取矩阵的构建提供具体方法。

3) 通过析取矩阵理论确定分组方案。以析取矩阵的列向量选择性为基础，根据检测对象的总数量以及其中包含非法者的数量确定析取矩阵的阶数，并由此构建分组检测方案，完成检测，根据分组检测结果推断非法数字签名在集合中的位置。

4) 对所提数字签名分组批量验证方案进行理论分析与仿真验证。理论分析证明了所提方案的正

确性、安全性、高效性，并以 Linux Ubantu20 为平台、Python3.9 为编程语言，引入双线性对函数库 pypbc，基于成熟的 BGLS 聚合签名算法[19]进行数字签名批量验证，仿真实现分组检测算法并识别非法对象。结果表明，所提方案能准确可靠地识别非法对象，同时在多个处理器并行计算时相较其他算法检测效率有所提高。

1 相关知识

本节介绍析取矩阵、区组设计、关联矩阵等非适应性分组检测的相关概念和背景知识。析取矩阵[12]用于非适应分组检测方案的非法数字签名的识别分析；区组设计[20]用于构造非适应性分组检测方案；关联矩阵[21]用于区组设计的矩阵化表示，是区组设计研究的一个有力的代数工具。

定义 1 d 阶析取矩阵[12]。对于 0-1 矩阵 $\boldsymbol{M}_{t\times n}$，若任意 d 列的并集不包含其他列向量，即 $\boldsymbol{C}_{j'} \not\subset \bigcup_{j\neq j'}^{d} \boldsymbol{C}_j$，则称 $\boldsymbol{M}_{t\times n}$ 为 d 阶析取矩阵。换句话讲，对于任意 d+1 列，必然存在至少一行（不妨设为第 i 行），使 d 列的元素 $m_{ij}=0$，即并集 $\bigvee_{j\neq j'}^{d} m_{ij}$=0，而 $m_{ij'}=1$。

定义 2 区组设计[20]。有限集合 X 上的任意一个子集族$\mathcal{B}$=$\{B_1,\cdots,B_t\}$ 为 X 上的一个区组设计，记作 $D=\{X,\mathcal{B}\}$。X 称为此设计的基集，而子集族$\mathcal{B}$中的诸子集 $B_i(i=1,\cdots,t)$ 则称为此设计的区组。

常见的区组设计包括成对平衡设计（PBD, pairwise balanced design）、可分组设计（GDD, group divisible design）、横截设计（TD, transversal design）、平衡不完全区组设计（BIBD, balanced incomplete block design）等[22]，本文主要应用横截设计进行区组设计。

定义 3 横截设计[22]。对于区组设计 $D=\{X,\mathcal{B}\}$，若有限集合 X 的一个划分为$\mathcal{G}$=$\{G_1,\cdots,G_k\}$（其中 $G_j, j=1,\cdots,k$ 称为组），且满足以下三点。

1) 对任意 $B\in\mathcal{B}$，$|B|=k$。

2) 对任意 $G_j\in\mathcal{G}$，$|G_j|=m$。

3) X 中属于同一组的不同元素在区组中的相遇数 $\lambda_D(x,y)$ 总为零，而属于不同组的 2 个元素 x,y 的相遇数 $\lambda_D(x,y)$ 是不依赖于 x,y 的常数，即对于任意 $x,y\in X$，$x\neq y$，有

$$\lambda_D(x,y)=\begin{cases}0, & x,y\text{属于划分的同一组}\\ \lambda, & x,y\text{属于划分的不同组}\end{cases}$$

则称其为横截设计，记作 TD[$k,\lambda;m$]。

定义 4 关联矩阵[21]。对于区组设计 $D=\{X,\mathcal{B}\}$，其中有限集合 $X=\{x_1,\cdots,x_n\}$ 及其子集族 $\mathcal{B}$= $\{B_1,\cdots,B_t\}$，存在 0-1 矩阵 $\boldsymbol{M}_{t\times n}=\{m_{ij}\}$，$i=1,\cdots,t$，$j=1,\cdots,n$，$t$ 表示区组个数，n 表示基集 X 元素个数，且

$$m_{ij}=\begin{cases}0, & x_j\notin B_i\\ 1, & x_j\in B_i\end{cases} \tag{1}$$

则称 $\boldsymbol{M}_{t\times n}$ 为区组设计的关联矩阵。

2 非适应性分组检测模型

根据分组检测对象数量进行区组设计，构建非适应性分组检测模型，关键在于寻找相应的析取矩阵，本文首先运用拉丁方理论进行区组设计——横截设计，然后证明构造的横截设计对应的关联矩阵是析取矩阵，最后再运用析取矩阵进行分组检测，完成非法对象的识别。

2.1 基于拉丁方理论的横截设计

横截设计是构造区组设计的重要方法之一，它与正交拉丁方理论有着密切关系。正交拉丁方是组合设计的一个重要研究课题，在横截设计的递归构造方法中，正交拉丁方组对于横截设计的存在性起着十分关键的作用。下面首先介绍正交拉丁方的一些概念和基本性质，然后论证它与横截设计的关系，最后给出正交拉丁方的构造方法，从而完成横截设计的构造。

定义 5 拉丁方[22]。设 S 是一个 m 元集，若 $\boldsymbol{A}$ 为 S 上的一个 $m\times m$ 阶阵列（即矩阵），其每一行与每一列都是集合 S 的一个全排列，则称 $\boldsymbol{A}$ 是 S 上的一个 m 阶拉丁方。

定义 6 正交拉丁方[22]。对于集合 S 和 S' 上的 2 个 m 阶拉丁方 $\boldsymbol{A}=(a_{ij})$ 和 $\boldsymbol{B}=(b_{ij})$，$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 在 (i,j) 位置上的有序对 $(a_{ij},b_{ij})\in S\times S'$ 称为一个对子，$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 在全部 m^2 个不同位置上的对子两两不同，即 $|R(\boldsymbol{A},\boldsymbol{B})|=m^2$，则称拉丁方 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 正交。若一组 m 阶拉丁方 $\boldsymbol{A}_1,\cdots,\boldsymbol{A}_t$ 两两正交，则称其为一个正交拉丁方组。

对于矩阵

$$\boldsymbol{H}=\begin{pmatrix}1 & \cdots & 1\\ 2 & \cdots & 2\\ \vdots & \ddots & \vdots\\ m & \cdots & m\end{pmatrix} \tag{2}$$

根据定义 6，显然 $\boldsymbol{H}$ 与其转置矩阵 $\boldsymbol{G}=\boldsymbol{H}^{\mathrm{T}}$ 正交。显然，若 $\boldsymbol{A}$ 是 $\{1,\cdots,m\}$ 上的任一 $m\times m$ 阶阵列，则 $\boldsymbol{A}$ 是 m 阶拉丁方的充要条件是 $\boldsymbol{A}$ 与 $\boldsymbol{H}$ 及 $\boldsymbol{A}$ 与 $\boldsymbol{G}$ 都正交。

定理 1[22]　存在 TD$[k,1;m]$的充要条件是存在一个 $k-2$ 个 m 阶正交拉丁方组。

证明　令 $D=(X,\mathcal{G},\mathcal{B})$是个 TD$[k,1;m]$，根据 TD 定义，则对任一 $B\in\mathcal{B}$及任一 $G_j\in\mathcal{G}$，有 $\left|B\cap G_j\right|=1$，即任一区组 B 由每个组中各取一个元素组成。任取划分$\mathcal{G}$的 2 个不同的组 G_1和 G_2（划分的每个组中含有 m 个元素），由于$\mathcal{B}$中的每个区组恰好包含了一个 G_1的元素和一个 G_2的元素，且由于 λ=1，G_1的元素和 G_2的元素在$\mathcal{B}$的每个区组中总共相遇一次，因此$\mathcal{B}$的区组数等于 G_1中所有元素和 G_2中所有元素的相遇总次数，即

$$\sum_{x\in G_1,y\in G_2}\lambda_D(x,y)=\lambda\left|G_1\right|\left|G_2\right|=m^2 \tag{3}$$

因此，$\mathcal{B}$中共有 m^2 个区组，G_1 的 m 个元素和 G_2 的 m 个元素在$\mathcal{B}$的每个区组中仅仅相遇一次，故 G_1 和 G_2 元素在区组设计$\mathcal{B}$中组成的有序数对 $\{(g_1,g_2)\,|\,g_1\in G_1,g_2\in G_2\}$ 仅仅出现一次可以看成 $\{1,\cdots,m\}$ 上 2 个正交的阵列，那么 $G_1,\cdots,G_k$ 的元素在区组设计$\mathcal{B}$中两两组成的有序数对仅仅出现一次可以看成 $\{1,\cdots,m\}$ 上两两正交的阵列，除了 $\boldsymbol{H}$ 和 $\boldsymbol{G}$，则一定还存在 $k-2$ 个正交拉丁方。

反之，若存在 $k-2$ 个 m 阶正交拉丁方，加上 $\boldsymbol{H}$ 和 $\boldsymbol{G}$，便可以构造 k 个两两正交的 m 阶阵列组，这些阵列两两之间每个元素对仅仅出现一次，照此便可以构造出一个 TD$[k,1;m]$。证毕。

然而，并不是任意阶的正交拉丁方都是存在的，Bose、Shrikhande 和 Parker 三人经过共同努力，证明了若 $n\neq 2,6$ 则必定存在一对拉丁方。由上述定理可知，在构造横截设计时，需要尽可能多的拉丁方。根据有限域存在定理[23]，对于任意素数幂 p^τ，存在一个含有 p^τ 个元素的有限域，显然 TD$[k,1;m]$ 的约束和有限域的特征非常相似，下面介绍通过有限域构造正交拉丁方组的方法。

定理 2　$m=p^\tau$ 素数幂阶拉丁方的构造。令 $m=p^\tau$ 是素数幂，$F=\{0,1,\cdots,m-1\}$，$F^*=F-\{0\}$，$\left|F\right|=m=p^\tau$，构造 $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(m-1)}$ 为 $\boldsymbol{A}^{(e)}=(a_{ij}^{(e)})$，其中 $a_{ij}^{(e)}=ei+j$（$1\leqslant e\leqslant m-1$，$0\leqslant i,j\leqslant m-1$），则 $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(m-1)}$ 是两两正交的拉丁方组。

证明　对于任一确定的 $1\leqslant e\leqslant m-1$，即 $e\in F^*$，若 i,j 分别取遍 F 中 m 个元素，则 $a_{ij}^{(e)}=ei+j$ 对应集合 F 上一个 $m\times m$ 阶矩阵 $\boldsymbol{A}^{(e)}=(a_{ij}^{(e)})_{m\times m}$；若 e 取遍 F^* 中 $m-1$ 个非零元素，则 $\boldsymbol{A}^{(e)}=(a_{ij}^{(e)})_{m\times m}$ 给出了集合 F 上的 $m-1$ 个 $m\times m$ 阶矩阵。

下面，首先证明每个 $\boldsymbol{A}^{(e)}$ 是 m 阶拉丁方，即每行（列）上的 m 个元素两两不同，假设 $\boldsymbol{A}^{(e)}$ 在第 i_1 行的 (i_1,j_1) 和 (i_1,j_2) 位置上，即 $a_{i_1j_1}^{(e)}=a_{i_1j_2}^{(e)}$，则 $ei_1+j_1=ei_1+j_2$，故 $j_1=j_2$；假设 $\boldsymbol{A}^{(e)}$ 在第 j_1 列的 (i_1,j_1) 和 (i_2,j_1) 位置上，即 $a_{i_1j_1}^{(e)}=a_{i_2j_1}^{(e)}$，则 $ei_1+j_1=ei_2+j_1$，故 $e(i_1-i_2)=0$，而 $e\neq 0$，则 $i_1=i_2$。

然后证明 $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(m-1)}$ 中任意 2 个拉丁方 $\boldsymbol{A}^{(e_1)}$ 和 $\boldsymbol{A}^{(e_2)}$ $(e_1\neq e_2)$ 正交，假设 $(a_{ij}^{(e_1)},a_{ij}^{(e_2)})=(a_{kl}^{(e_1)},a_{kl}^{(e_2)})$，则 $a_{ij}^{(e_1)}=a_{kl}^{(e_1)},a_{ij}^{(e_2)}=a_{kl}^{(e_2)}$，即

$$\begin{cases}e_1i+j=e_1k+l\\ e_2i+j=e_2k+l\end{cases} \tag{4}$$

则 $(e_1-e_2)i=(e_1-e_2)k$。

由于 $e_1\neq e_2$，故 $(e_1-e_2)^{-1}$，因此 $i=k$，代入可得 $j=l$。

综上所述，$\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(m-1)}$ 是两两正交的拉丁方组。证毕。

2.2　基于横截设计的析取矩阵

析取矩阵是一类二元叠加码，可以用作分组检测的一种数学模型，广泛应用在信息传输中的多路存取信道、分子生物学、基因遗传测试、病毒分组检测等诸多方面，横截设计 TD$[k,1;m]$关联矩阵 $\boldsymbol{M}$ 的转置矩阵具备析取矩阵的特征，以下将从理论上证明这种推断。

在 t 行 n 列的 0-1 矩阵 $\boldsymbol{M}$ 中，$\boldsymbol{C}_j$ 表示第 j 列向量；w_j 表示第 j 列向量 $\boldsymbol{C}_j$ 的重量，即 $\boldsymbol{C}_j$ 中“1”的个数；λ_{ij} 表示列向量 $\boldsymbol{C}_i$ 与 $\boldsymbol{C}_j$ 的点乘，即 $\boldsymbol{C}_i$ 与 $\boldsymbol{C}_j$ 相同行上都为 1 的个数，也称为 $\boldsymbol{C}_i$ 与 $\boldsymbol{C}_j$ 相交 λ_{ij} 次。

引理 1[12]　$\boldsymbol{M}$ 是 $t\times n$ 矩阵，其中列向量重量最

小值为 $\underline{w}=\min\limits_{j} w_j$，任意两列内积最大值为 $\overline{\lambda}=\max\limits_{i,j}\lambda_{ij}$，则 $\boldsymbol{M}$ 是 d 阶析取矩阵，其中 $d=\left\lfloor\dfrac{\underline{w}-1}{\overline{\lambda}}\right\rfloor$。

证明 根据 $\overline{\lambda}$ 的定义可知，矩阵 $\boldsymbol{M}$ 中任意 2 个列向量相交次数最大为 $\overline{\lambda}$，因此列向量 $\boldsymbol{C}_j$ 与不包含 $\boldsymbol{C}_j$ 的 d 个列向量的并集相交的次数最大为 $d\overline{\lambda}$。如果令 $d=\left\lfloor\dfrac{\underline{w}-1}{\overline{\lambda}}\right\rfloor$，则

$$d\overline{\lambda}=\left\lfloor\frac{\underline{w}-1}{\overline{\lambda}}\right\rfloor\overline{\lambda}<\lfloor(\underline{w}-1)\rfloor<\underline{w} \tag{5}$$

又因为 $\underline{w}<w_j$，则 $d\overline{\lambda}<\underline{w}<w_j$，表示 $\boldsymbol{C}_j$ 与不包含 $\boldsymbol{C}_j$ 的 d 个列向量的并集相交的最大次数 $d\overline{\lambda}$ 小于 $\boldsymbol{C}_j$ 的重量 w_j，因此 $\boldsymbol{C}_j$ 不能被这样的 d 个列向量的并集所包含，故 $\boldsymbol{M}$ 是 d 阶析取矩阵。证毕。

定理 3 当 TD$[k,\lambda;m]$设计的 $\lambda=1$ 时，其关联矩阵 $\boldsymbol{M}$ 的转置矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 的 $\lambda^{\mathrm{T}}=0$ 或 1，则 $\boldsymbol{M}^{\mathrm{T}}$ 是 $k-1$ 阶析取矩阵。

证明 当 $\lambda=1$ 时，TD 区组设计对应关联矩阵 $\boldsymbol{M}$ 的任意两列的内积为 1，即 $\boldsymbol{M}$ 的任意两列仅仅有一次在相同位置（行）都为“1”，则 $\boldsymbol{M}$ 的任意两行元素仅仅有一次在相同位置（列）都为“1”，下面用反证法进行证明。假设 $\boldsymbol{M}$ 的任意两行元素至少有两次在相同位置（列）都为“1”，此时 $\boldsymbol{M}$ 的任意两列元素至少有两次在相同位置（行）都为“1”，与 $\lambda=1$ 相矛盾，故 $\boldsymbol{M}$ 的转置矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 的 $\lambda^{\mathrm{T}}=0$ 或 1，则 $\overline{\lambda}^{\mathrm{T}}=1$，此时 $\underline{w}$=k，由引理 1 可知 $\boldsymbol{M}^{\mathrm{T}}$ 是 $k-1$（即 $\underline{w}-1$）阶析取矩阵。证毕。

2.3 基于析取矩阵的分组检测

采用分组检测进行数字签名验证，将数字签名进行分组，分组结果用关联矩阵 $\boldsymbol{M}_{t\times n}$ 表示，其中矩阵的列对应要检验的数字签名，矩阵的行对应分组，$\boldsymbol{M}_{t\times n}$ 中元素 $\{m_{ij}\}$ 的取值参考定义 4。

不妨假设将要进行分组批量验证的数字签名集合 $\Sigma=(\sigma_1,\cdots,\sigma_n)$ 的状态为 $X=(x_1,\cdots,x_n)$，其中 $x_i(i=1,\cdots,n)$ 表示每个数字签名的实际状态，$x_i=0$ 表示第 i 个数字签名合法，$x_i=1$ 表示第 i 个数字签名非法。为了快速进行数字签名验证，运用 2.1 节的横截设计对数字签名进行分组，并行进行分组检测，用 $\boldsymbol{Y}$ 表示分组检测结果，记为 $\boldsymbol{Y}=(y_1,\cdots,y_t)$，其中 $y_j=0$ $(j=1,\cdots,t)$ 表示本组检测的数字签名集合全部合法，$y_j=1$ $(j=1,\cdots,t)$ 表示本组检测的数字签名集合至少包含一个非法数字签名，它与 X 和 $\boldsymbol{M}_{t\times n}$ 的关系可以表示为

$$\boldsymbol{Y}=\boldsymbol{M}_{t\times n}X \tag{6}$$

根据 2.2 节的证明，上述分组批量检测区组设计的关联矩阵 $\boldsymbol{M}_{t\times n}$ 为 d 阶析取矩阵，当数字签名集合 X 中非法数字签名的个数不超过 d 时，运用算法 1 便可以确定非法数字签名。

算法 1 非法数字签名分组检测法

输入 $\boldsymbol{Y}=(y_1,y_2,\cdots,y_{km})$

输出 非法数字签名集合 U

```
V = ∅
for y_i in Y
  if y_i == 0
    for j in range(0,n)
      if m_ij == 1
        V = V ∪ X_j
      end if
    end for
  end if
U = X \ V
end for
```

首先输入 d 阶析取矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 及分组检测结果 km 维 0-1 向量 $\boldsymbol{Y}=(y_1,y_2,\cdots,y_{km})$，根据 $\boldsymbol{Y}$ 中“0”逐行检查 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 中所对应列，并加入合法数字签名集合 V。其次使用排除法得到非法数字签名 $U=X\setminus V$，具体如算法 1 所示。最后若 $|U|\leqslant d$，确定非法数字签名；否则，需要对 U 中数字签名进行逐一认证确定非法数字签名。

3 数字签名分组检测实施

网络的中心节点每秒接收到大量节点发来的消息，为加快消息签名的验证速度，通常采用“聚合签名+分组检测”的方法进行数字签名的验证，如图 2 所示。首先采用聚合签名算法将所有来自不同节点的消息签名设计为具有同态性特征，便于后续聚合验证作为中心节点的信息处理中心先将所有消息及其签名进行一次聚合验证，若验证成功，则表明所有消息合法；否则，证明消息集合中至少存在一个非法对象，需要进一步采取非适应性分组的思想进行分组聚合验证，用 i 表示分组序号，i

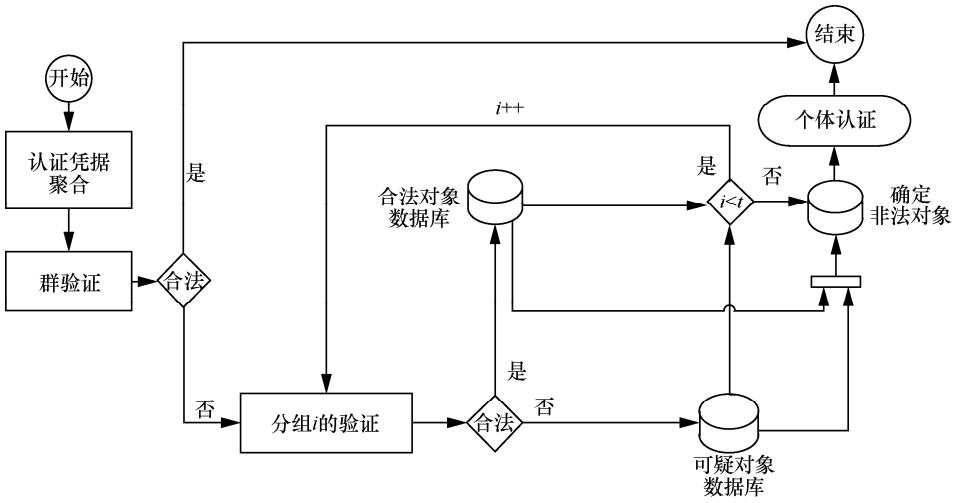


图 2　数字签名快速验证方法

取遍 t 个分组，通过分组验证的结果排除合法对象，从而确定非法对象，当出现无法确定的对象时还可以补充进行个体认证。由此可见，实现消息签名批量验证的是聚合签名算法，识别非法对象依靠分组检测，不同的分组方案对应着不同的识别效率。

3.1　聚合签名

为提高数字签名的验证效率，在一般数字签名算法的基础上，杨涛等[8]提出一种具有同态性结构的变体数字签名算法，验证者能够将来自任意不同节点的多个数字签名压缩合成为与单个数字签名几乎同等大小的短签名进行批量验证，大大减小了多个签名验证的工作量，这就是聚合签名。聚合签名由于其高效性、简捷性，广泛应用于安全路由协议、网云信息聚合、日志审计、分布式计算等方面，在诸多信息科技领域发挥着重要的安全保护作用，成为近年来被关注的一个研究热点。本文采用经典的 BGLS 聚合签名算法[19]完成消息签名的批量验证，下面对该算法进行阐述。

设 G_1 和 G_2 是 2 个 p 阶的乘法循环群（p 为素数），g_1 和 g_2 分别是 G_1 和 G_2 的生成元；$\psi: G_1 \to G_2$ 是一个同构映射，且 $\psi(g_1)=g_2$；$e: G_1 \times G_2 \to G_T$ 是一个双线性对运算；$h: \{0,1\}^* \to G_2$ 是一个杂凑函数，可以看作一个随机预言机。

密钥生成。选择随机数 $x \in_R Z_p$，计算 $v=g_1^{\,x}$，则 $v \in G_1$ 作为用户的公钥，$x \in Z_p$ 作为用户的私钥。

签名生成。对于用户 $u_i \in U$，公钥为 v_i，私钥为 x_i，需要签名的消息为 $M_i \in \{0,1\}^*$，可得 $h_i=h(M_i)$，$\sigma_i=h_i^{x_i}$。

聚合验证。设 k 个用户分别对 k 个消息 M_i（$1 \leqslant i \leqslant k$）生成的签名为 $\{\sigma_1,\cdots,\sigma_k\}$，将这些签名聚合起来是 $\sigma=\prod_{i=1}^{k}\sigma_i$，$\sigma \in G_2$ 便是聚合签名；使用 k 个用户的公钥 $v_i \in G_1$ 和消息 M_i（其中 $1 \leqslant i \leqslant k$）验证 σ 便可以完成 $\{\sigma_1,\cdots,\sigma_k\}$ 的验证，具体验证式为

$$e(g_1,\sigma)=\prod_{i=1}^{k}e(v_i,h_i) \tag{7}$$

BGLS 聚合签名算法的安全性等价于随机预言模型下 CDH（computational Diffie-Hellman）问题的安全性。

3.2　实施步骤

假设检测 $n=m^2$（m 为素数或素数幂）个数字签名集合，其中含有不超过 $d=k-1$ 个非法数字签名，根据拉丁方理论便可以构造一个横截设计 TD$[k,1;m]$，当 $m>k$ 时，通过 km 次分组检测完成 m^2 个数字签名的验证，识别出非法数字签名，如图 3 所示，具体步骤如下。

步骤 1　确定分组检测参数设置。根据需要检测的数字签名集合大小 n 以及非法对象数量上限 d，确定横截设计 TD$[k,1;m]$的参数 (k,m)，如果数字签名的总数及非法对象的个数不满足上述要求，可以进行适当的冗余填充，达到要求。

步骤 2　构造 $k-2$ 个正交拉丁方。采用定理 2 构造 $k-2$ 个 m 阶正交拉丁方 $\boldsymbol{A}^{(1)},\cdots,\boldsymbol{A}^{(k-2)}$。

步骤 3　由 $\boldsymbol{H},\boldsymbol{G},\boldsymbol{A}^{(1)},\cdots,\boldsymbol{A}^{(k-2)}$ 构造 TD$[k,1;m]$。

步骤 4　编写 TD$[k,1;m]$的关联矩阵 $\boldsymbol{M}_{m^2\times km}$，求出 d 阶析取矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$。

步骤 5 按照 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 进行分组检测，通过检测结果由算法 1 推知非法数字签名。

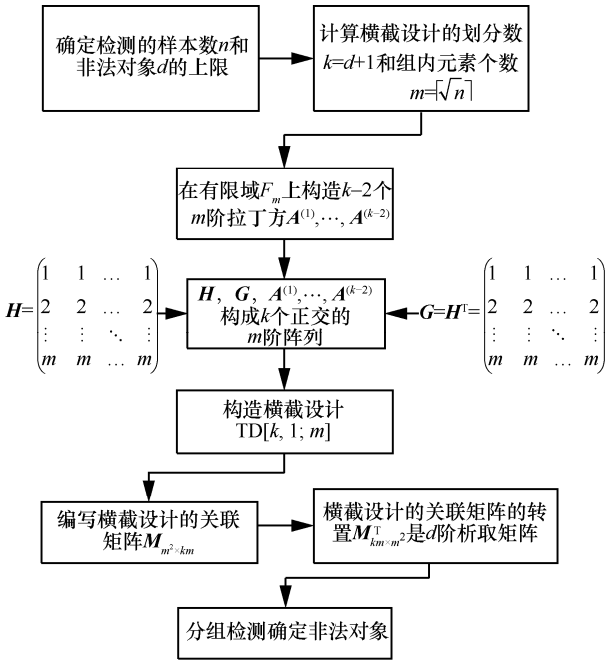


图 3 数字签名分组检测实施步骤

3.3 举例说明

如果对 25 个数字签名进行验证，据前期统计估计其中包含非法数字签名的数量不超过 2 个，则令 $n=m^2=5^2$，$d=k-1=2$，则 $k=3$，$m=5$，确定横截设计为 TD[3,1;5]。

利用定理 2 构造一个 5 阶拉丁方矩阵

$$\begin{pmatrix}1&2&3&4&5\\2&3&4&5&1\\3&4&5&1&2\\4&5&1&2&3\\5&1&2&3&4\end{pmatrix} \tag{8}$$

连同 $\boldsymbol{H}$ 和 $\boldsymbol{G}$，按照定理 1 的方法，由 3 个拉丁方一起构成一个横截设计 TD[3,1;5]，其关联矩阵为 25 行 15 列矩阵 $\boldsymbol{M}_{25\times 15}$，转置后得到 15 行 25 列 0-1 矩阵 $\boldsymbol{M}^{\mathrm{T}}_{15\times 25}$，其中每列有 3 个“1”，每行有 5 个“1”，如图 4 所示。

根据定理 3 可知，0-1 矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 为 2 阶析取矩阵。图 4 中的矩阵 25 列表示需要验证的 25 个数字签名，15 行表示 25 个数字签名分成的 15 个分组，矩阵中 (i,j) 处的元素“1”表示 j 列的数字签名属于 i 行的分组；否则，表示不在此分组。析取矩阵的性质表明，按照图 4 中分组方案进行批量验证，可以通过 15 次分组检测对 25 个数字签名完成不超过 2 个非法对象的识别。

为了说明分组方案在数字签名批量验证中的使用思路，不妨假设第 3、6 个数字签名是非法的，在图 4 中用方框表示，按照分组方案调用聚合签名算法完成 15 个分组的批量验证，得到检测结果 $\boldsymbol{Y}=(1,1,0,0,0,1,0,1,0,0,0,1,1,0,0)^{\mathrm{T}}$，由于 $\boldsymbol{M}_{15\times 25}$ 是 2 阶析取矩阵，$\boldsymbol{Y}$ 中的“0”表示本组检测中所有数字签名均为合法，根据算法 1，采用排除法可以依次确定合法数字签名 x_{11}、x_{12}、x_{13}、x_{14}、x_{15}、x_{16}、

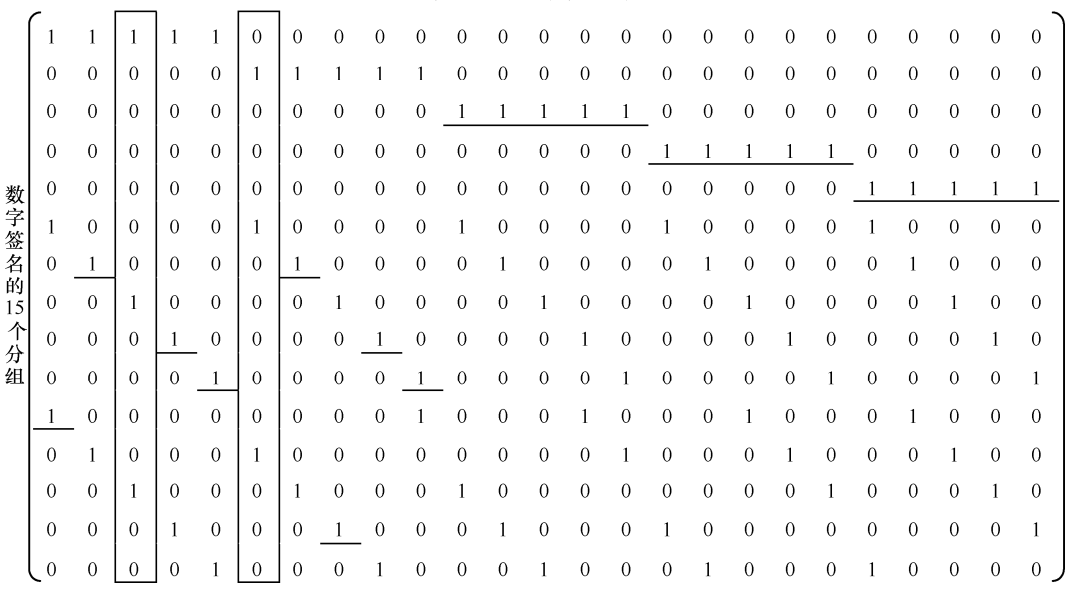


图 4 15 个数字签名的分组验证的分组情况

x_{17}、x_{18}、x_{19}、x_{20}、x_{21}、x_{22}、x_{23}、x_{24}、x_{25}、x_2、x_7、x_4、x_9、x_5、x_{10}、x_1、x_8，如图 4 中下划线所示；根据析取矩阵的性质可知，剩下的签名 x_3、x_6 便为非法数字签名。

4　性能分析

本节通过理论证明和仿真实验，对本文算法进行了可行性和复杂度分析，并对非法数字签名数量估计不准的情况讨论了方案的容错性；同时借助开源的双线性对计算函数库，编程实现了 BGLS 聚合签名方案和本文算法，仿真验证了数字签名分组检测方案。结果表明，在相同安全要求下，本文算法相较于逐一验证具有高效性，即使与经典的二分法检测相比也具有明显的效率优势。

4.1　理论证明

1) 可行性分析

根据算法 1 的排除法可知，本文算法是通过分组检测结果中合法结果（即检测结果为“0”分组）选出合法数字签名，然后取合法数字签名对应的余集作为可疑签名集 $U = X \setminus V$，若 $|U| \leqslant d$ 则确定非法数字签名。之所以能够识别非法数字签名，是因为关联矩阵的析取性，下面根据析取矩阵的特性，对其进行证明。

定理 4　群组认证关联矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 是 d 阶析取矩阵，则由分组检测的结果 Y 通过算法 1 可以识别不超过 d 个非法数字签名。

证明　已知有 $n = m^2$ 个数字签名要进行检测，其中非法数字签名数量上限不超过 d，不妨设 $x_{j_1},\cdots,x_{j_d}$ 是非法者，x_j 为任意一个合法数字签名，则经过关联矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 分组检测得到 $\boldsymbol{Y} = \boldsymbol{M}^{\mathrm{T}}_{km\times m^2}\boldsymbol{X}$，其结果 $\boldsymbol{Y}$ 便为关联矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 中 $x_{i1},\cdots x_{id}$ 相应列的布尔和（并），$\overset{d}{\underset{k=1}{\vee}} x_{j_k}$。由于矩阵 $\boldsymbol{M}^{\mathrm{T}}_{km\times m^2}$ 是 d 阶析取矩阵，合法数字签名 x_j 对应的列不在 $x_{j_1},\cdots,x_{j_d}$ 相应列的布尔和（并）中，$x_j \notin \overset{d}{\underset{k=1}{\vee}} x_{j_k}$，即存在某行 $m_{ij_k} = 0$，$(k = 1,\cdots,d)$，而 $m_{ij} = 1$，这样 x_j 便可以被识别。证毕。

2) 复杂度分析

本文算法对 n 个数字签名进行验证，初步估计非法数字签名的数量不超过 d 个。下面对本文算法的复杂度进行分析。

定理 5　本文算法进行分组检测，通过 $\lceil (d+1)\sqrt{n} \rceil$ 个分组完成 n 个数字签名进行验证，非法数字签名识别算法（算法 1）的时间复杂度为 $O(\sqrt{n})$，其中 d 为非法数字签名的数量上限，且 $d \ll n$。

证明　要完成 n 个数字签名进行验证，根据 3.2 节中的实施步骤，首先基于拉丁方构造了用于分组检测的析取矩阵 $\boldsymbol{M}^{\mathrm{T}}$，它的行数为 km，列数为 m^2，由于 $k = d+1$，$n = m^2$，则

$$km = \lceil (d+1)\sqrt{n} \rceil \tag{9}$$

另外，当分组批量验证完成后，需要通过算法 1 运用排除法完成非法数字签名的识别，只有分组批量验证为“0”的分组才参与排除法，不妨用 t_0 表示批量验证 $\boldsymbol{Y} = (y_1, y_2, \cdots, y_{km})$ 中为“0”的分组，根据横截设计的“0”“1”分布特征，关联矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 的每列元素“1”的个数为 k，分组批量验证结果 $\boldsymbol{Y} = (y_1, y_2, \cdots, y_{km})$ 便为 $\boldsymbol{M}^{\mathrm{T}}$ 的所有非法数字签名对应列的布尔和，因此结合非法数字签名的数量上限 d，可知 $\boldsymbol{Y}$ 中“0”的个数满足

$$k(m-d) < t_0 < k(m-1) \tag{10}$$

又因为 $n = m^2$，$k = d+1$，则 $t_0 = \lceil (d+1)\xi \rceil$，其中 $\sqrt{n} - d < \xi < \sqrt{n} - 1$。通常情况下 $d \ll n$，故 t_0 的复杂度为 $O(\sqrt{n})$。

3) 容错性分析

容错性是指当非法数字签名数量估计不准确时，分组检测方案识别非法数字签名的能力变化情况。显然当非法数字签名的实际数量 $d' \leqslant d$ 时，关联矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 为 d 阶析取矩阵的分组检测方案识别能力没有变化，因此重点讨论 $d' > d$ 的情况，此时分组验证的结果 $\boldsymbol{Y} = (y_1, y_2, \cdots, y_{km})$ 便为所有 d' 列元素的布尔和。因为关联矩阵 $\boldsymbol{M}^{\mathrm{T}}$ 为 d 阶析取矩阵，所以无法通过算法 1 准确识别非法数字签名，却可以确定一个包含所有非法数字签名的更大的可疑对象集合，此时可以采用图 1 的方法对可疑对象进行逐一验证，从而完成非法数字签名的识别。具体举例如下，参照图 4 的关联矩阵，需要验证的数字签名个数仍为 25，初步估计其中非法者的个数不超过 2 个，但实际第 3、11、17 个数字签名是非法的，利用上述分组检测得到检测结果 $\boldsymbol{Y} = (1,0,1,1,0,1,1,1,0,0,0,0,0,1,0,1)^{\mathrm{T}}$，由于 $\boldsymbol{M}_{15\times 25}$ 是 2 阶析取矩阵，$\boldsymbol{Y}$ 中的“0”表示本组检测中所有数字签名均为合法，采用排除法可以依次确定合法数字签

名 x_6、x_7、x_8、x_9、x_{10}、x_{21}、x_{22}、x_{23}、x_{24}、x_{25}、x_4、x_{14}、x_{19}、x_5、x_{15}、x_{20}、x_1、x_{18}、x_2、x_6、x_{12}、x_{16}，剩下的数字签名 x_3、x_{11}、x_{13}、x_{17} 都为可疑对象；此时剩下的数字签名数量超过 3，证明非法数字签名数量估计偏少，2 阶析取的关联矩阵无法完成非法数字签名的确定。如果剩余签名数量和初始估计的签名数量相差较大，则需要重新设计分组检测方案；否则，便可以逐一认证剩下的签名，确定非法对象。因此，逐一验证 x_3、x_{11}、x_{13}、x_{17} 便可以完成所有数字签名的验证。

4.2 仿真比较

为检验本文算法的实际性能，在处理器为 Intel Core i5-4200 2.5 GHz、内存为 4 GB 的笔记本电脑的虚拟机 VMware Workstation Pro 上安装 Linux Ubantu20 操作系统，在 Python 3.9 编程环境下基于 GMP（GNU multiple precision）和 PBC（pairing-based cryptography）配置 pypbc 双线性对匹配加密库，杂凑函数采用 SHA-256，仿真环境配置如图 5 所示。

```
from pypbc import *
import hashlib, time
Hash2 = hashlib.sha256
#public parameters is globle or may be "segmentation fault(core dumped)"
params=Parameters(qbits=512, rbits=160)
pairing=Pairing(params)
g_1=Element.random(pairing, G1)#g is the generator of G1
```

图 5 仿真环境配置

基于 BGLS 聚合签名方案进行数字签名的批量处理，以数字签名的数量 n 及其所含非法数字签名的上限 d 作为分组的依据，采用 3.2 节的步骤进行分组检测，使用算法 1 进行非法对象的识别，部分仿真结果如表 1 所示。

表 1 数字签名分组检测部分仿真结果

序号	n	t	k	m	d	时间/s
1	49	21	3	7	2	1.886
2	361	76	4	19	3	4.166
3	961	155	5	31	4	9.420
4	1 849	301	7	43	6	21.988
5	4 096	640	10	64	9	36.560
6	4 489	737	11	67	10	39.211
7	4 489	1 541	23	67	22	90.660
8	4 489	2 077	31	67	30	132.267
9	4 489	3 149	47	67	46	198.886
10	4 489	3 685	55	67	54	223.980

分组检测的完成时间主要与分组检测次数相关，当数字签名的数量、非法数字签名的数量增加时，分组个数必然增加，如图 6 所示。随着分组检测次数的增加，完成数字签名验证的时间也在增加，正如表 1 所示。另外，从仿真实验中发现，当数字签名的总数 n 不变，但非法数字签名的数量增加到接近 $n/3$ 时，分组检测的效果低于逐一验证的效率，与文献[12]结论一致。

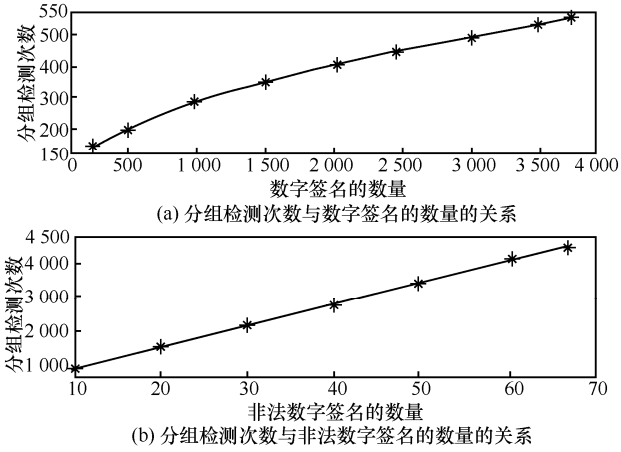


图 6 分组检测次数与数字签名的数量、非法数字签名的数量的关系

非适应性分组检测方案与序贯性分组检测方案比较，一个重要区别在于分组平行处理，本文仿真采用 Thread 函数进行平行检测处理，如表 2 所示。当数字签名的数量和非法数字签名的数量分别为 n 和 d 时，列出逐一验证、二分法验证和本文算法在不同处理器数量时的最大检测数量。逐一验证需要对每个签名进行一次验证，验证次数为 n，在 p 个处理器工作的情况下，可以将数字签名分为 p 个分组，需要 $\left\lceil \frac{n}{p} \right\rceil$ 次检测；二分法验证采用折半查找法，每次需要 $\lceil \log n \rceil$ 次检验，只能检测一个非法数字签名，d 个非法数字签名需要 $d\lceil \log n \rceil$ 次检验，在 p 个处理器工作的情况下，可以将数字签名分为 p 个分组，分别进行折半查找，需要 $d\left\lceil \log \frac{n}{p} \right\rceil$ 次检测；本文算法通过 $\left\lceil (d+1)\sqrt{n} \right\rceil$ 个分组完成 n 个数字签名中 d 个非法数字签名识别，在 p 个处理器工作的情况下，可以将数字签名分为 $\frac{(d+1)\sqrt{n}}{p}$ 个分组同时进行，需要 $\frac{(d+1)\sqrt{n}}{p}$ 次验证。

本文算法采用并行处理的设计思想，为便于验

证的同时进行处理，设计了$\left\lceil (d+1)\sqrt{n} \right\rceil$个独立并行的分组进行验证数字签名。从表 2 可以看出，单个处理器时二分法验证效率最高，达到 $O(\log n)$，而本文算法的 $O(\sqrt{n})$仅比逐一验证的 $O(n)$较好一些。当多个处理器同时处理时，所有算法都可以并行处理，不管是逐一验证、二分法验证还是本文算法的检验效率都有所提高，本文算法的效率提升明显，提高到单个处理器时的 p 倍，而二分法验证提高了 $\log_n p$ 倍，显然 $p > \log_n p$（通常 $n > p$）。

表 2　　数字签名分组检测比较

算法	最大检测次数	
	单个处理器	p 个处理器
逐一验证	n	$\left\lceil \frac{n}{p} \right\rceil$
二分法验证	$d\lceil \log n \rceil$	$d\left\lceil \log \frac{n}{p} \right\rceil$
本文算法	$(d+1)\sqrt{n}$	$\frac{(d+1)\sqrt{n}}{p}$

如图 7 所示，将本文算法与逐一验证和二分法验证进行比较发现，当单个处理器串行验证时，二分法验证有相对较好的表现，同等数量的数字签名需要的验证次数最少；当多个处理器并行验证时，在非法数字签名数量 d=8、处理器个数 p=100 的情况下，虽然逐一验证和二分法验证可以用多个处理器同时进行验证，效率有所提升，但是随着数字签名的数量增多，本文算法表现出较强的效率优势。当数字签名的总数大于 3 000 时，本文算法与逐一验证相比才体现出优势，在多处理器保障的条件下，本文算法分组检测的次数随着数字签名总数的变化不大，基本上保持在 8 附近，与二分法验证、逐一验证相比，具有最少的验证次数，效率最高。随着数字签名的数量不断增大，尤其是$\frac{n}{d}$的增大，当多处理器并行验证时本文算法会表现出更多优势。

5　结束语

数字签名的非适应性分组验证是为提高网络中心节点数字签名的验证效率，将组合设计理论与聚合签名理论相结合，以横截设计理论为基础设计数字签名分组方案，以聚合签名理论为方法完成数字签名的批量分组验证，实现数字签名的并行快速验证。它通常应用于传感器网、交通网、物联网等具有海量节点信息需要安全验证，且消息时敏性强的无线网络环境。本文基于组合数学的拉丁方理论设计了数字签名分组方案，并以经典的 BGLS 聚合签名理论进行数字签名的批量验证，构建了一个以 $O(\sqrt{n})$次数完成 n 个数字签名验证的非适应性数字签名分组验证方案。该方案与逐一认证相比具有较高验证效率；在多处理器保障的条件下，与序贯类分组方案相比具有验证次数少、效率高的优势。但本文研究未考虑数字签名在信道中的传输错误问题，需要进一步研究加入检错纠错机制，增强分组检测的容错性能。

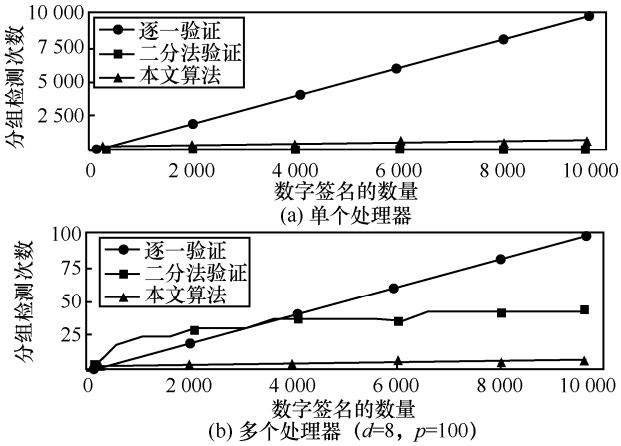


图 7　分组检测次数与数字签名的数量的关系

参考文献：

[1] ZHANG C X, HO P H, TAPOLCAI J. On batch verification with group testing for vehicular communications[J]. Wireless Networks, 2011, 17(8): 1851-1865.

[2] WANG Y H. A trust management model for Internet of vehicles[C]//Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy. New York: ACM Press, 2020: 136-140.

[3] 王宏, 李建华, 赖成喆, 等. 基于纠错码理论的群组认证[J]. 电子学报, 2019, 47(7): 1393-1400.
WANG H, LI J H, LAI C Z, et al. Group authentication based on error correction coding theory[J]. Acta Electronica Sinica, 2019, 47(7): 1393-1400.

[4] MAKAROV A. A survey of aggregate signature applications[J]. Advanced Technologies in Robotics and Intelligent Systems, 2020, 80(1): 309-317.

[5] KOZINA G L, SAVCHENKO D K. Aggregate signature protocol with group leader[J]. Cybernetics and Systems Analysis, 2021, 57(1): 165-172.

[6] TEZUKA M, TANAKA K. Improved security proof for the camenisch-lysyanskaya signature-based synchronized aggregate signature scheme[M]. Cham: Springer, 2020.

[7] HE M, LI X M, NI J B, et al. Balancing efficiency and security for network access control in space-air-ground integrated networks[C]//Proceedings of 2021 18th International Conference on Privacy, Security and Trust (PST). Piscataway: IEEE Press, 2021: 1-10.

[8] 杨涛, 孔令波, 胡建斌, 等. 聚合签名及其应用研究综述[J]. 计算机研究与发展, 2012, 49(S2): 192-199.
YANG T, KONG L B, HU J B, et al. Survey on aggregate signature and its applications[J]. Journal of Computer Research and Development, 2012, 49(S2): 192-199.

[9] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[J]. IACR Cryptology ePrint Archive, 2002, 2002: 175.

[10] HWANG J Y, LEE D H, YUNG M. Universal forgery of the identity-based sequential aggregate signature scheme[C]//Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. [S.l.:s.n.], 2009: 157-160.

[11] WEI X J, ZHANG L, LU D J. An aggregate signature scheme with forward security and non-repudiation[C]//Proceedings of the 2019 2nd International Conference on Information Hiding and Image Processing. New York: ACM Press, 2019: 15-20.

[12] DU D Z, HWANG F K. Combinatorial group testing and its applications[M]. Singapore: World Scientific, 1993.

[13] ZAVERUCHA G M, STINSON D R. Group testing and batch verification[J]. IACR Cryptology ePrint Archive, 2009, 2009: 240.

[14] SCARLETT J, CEVHER V. Phase transitions in group testing[C]// Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2016: 40-53.

[15] SHANGGUAN C, GE G N. New bounds on the number of tests for disjunct matrices[J]. IEEE Transactions on Information Theory, 2016, 62(12): 7518-7521.

[16] INDYK P, NGO H Q, RUDRA A. Efficiently decodable non-adaptive group testing[C]//Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2010: 1126-1142.

[17] ISCEN A, FURON T. Group testing for identification with privacy[C]//Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security. New York: ACM Press, 2016: 51-56.

[18] CHERAGHCHI M, NAKOS V. Combinatorial group testing and sparse recovery schemes with near-optimal decoding time[C]// Proceedings of 2020 IEEE 61st Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press, 2020: 1203-1213.

[19] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[C]//Advances in Cryptology — EUROCRYPT 2003. [S.l.:s.n.], 2003: 416-428.

[20] BRUALDI R. Introductory combinatorics[M]. New Jersey: Prentice Hall, 2004.

[21] PASTUSZAK J, PIEPRZYK J, SEBERRY J. Codes identifying bad signatures in batches[C]//Proceedings of 2000 International Conference on Cryptology in India (INDOCRYPT). Berlin: Springer, 2000: 143-154.

[22] 邵嘉裕. 组合数学[M]. 上海: 同济大学出版社, 1991.
SHAO J Y. Combinatorial mathematics[M]. Shanghai: Tongji University Press, 1991.

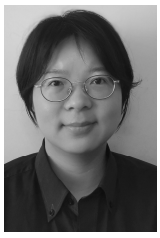
[23] WAN Z X. Finite fields and galois rings[M]. Singapore: World Scientific Publishing, 2011.

[作者简介]

王宏（1979− ），男，陕西澄城人，博士，国防科技大学讲师，主要研究方向为无线网络安全、有限域等。

赖成喆（1985− ），男，陕西汉中人，博士，西安邮电大学教授、硕士生导师，主要研究方向为无线网络安全。

刘向阳（1976− ），男，河南许昌人，博士，国防科技大学副教授、硕士生导师，主要研究方向为无线传感器网络、信号检测等。



曾晗（1991− ），女，河南南阳人，国防科技大学讲师，主要研究方向为无线网络安全。