ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 38,No. 2,Mar. 2023,pp. 413－425
DOI:10. 16337/j. 1004-9037. 2023. 02. 016

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于可再分发抗合谋编码的低失真水印方案

邸云龙，张迎周，汪天琦，李鼎文，朱林林

（南京邮电大学计算机科学学院，南京 210023）

摘　要：在大数据的时代背景下，数据潜在的价值使其成为重要的财富之一。数据的非法篡改、修正和非法分发给追踪数据的泄密源头带来巨大挑战。数字指纹技术可应用在数据泄密溯源领域，即在数据中嵌入一串能够唯一标识用户信息的序列，当数据发生泄密后，提取其中蕴含的数字指纹，追踪到泄密的叛逆者。多个用户对数据合谋攻击并泄密数据，从而毁坏其中嵌入的指纹信息达到逃脱追责的目的，抗合谋编码能够解决此问题。针对现有的数字指纹编码无法满足数据再分发需求、数字指纹嵌入造成较大数据失真的问题，本文使用BIBD（Balanced incomplete block design）作为外码，码字扩展后的C码作为内码，构建一种可再分发的抗合谋指纹编码RD-ACC（Redistributable anti-collusion fingerprint coding）。在此基础上，提出一种基于多目标优化的数据库指纹算法，能够在较小的数据库失真情况下保证数字指纹较高的鲁棒性，提取出的RD-ACC能够有效抵抗组内、组间多用户合谋攻击。实验结果表明，该算法能够在较小的数据失真下实现数据的再分发操作，并且能抵抗合谋攻击进行泄密溯源。

关键词：抗合谋数字指纹；数字水印；优化模型；数据泄密溯源；数据分发

中图分类号：TP391　　**文献标志码：**A

Low-Distortion Watermark Scheme Based on Redistributable Anti-collusion Coding

DI Yunlong, ZHANG Yingzhou, WANG Tianqi, LI Dingwen, ZHU Linlin
(College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: In the era of big data, the potential value of data makes it one of the important assets. The illegal tampering, correction and illegal distribution of data bring great challenges to tracing the source of data leakage. Digital fingerprint technology can be applied in the field of traceability of data leakage, that is, a sequence of unique identification of user information is embedded in the data. Multiple users conspire to attack data and leak data, thereby destroying the fingerprint information embedded in it to escape accountability. Anti-collusion coding can solve this problem. Aiming at the problems that the existing digital fingerprint encoding cannot meet the data redistribution requirements and the digital fingerprint embedding causes large data distortion, this paper uses the balanced incomplete block design (BIBD) as the outer code and the C code after codeword expansion as the inner code to construct a redistribution anti-collusion fingerprint coding (RD-ACC). On this basis, a database fingerprint algorithm based on multi-objective optimization is proposed to ensure high robustness of digital fingerprints under the condition of small database distortion. The extracted RD-ACC can effectively resist intra-group and inter-group multi-user collusion attack. Experimental results show that the algorithm can realize the data redistribution operation with less data distortion, and resist the collusion attack to trace the source of leaks.

Key words: anti-collusion digital fingerprint; digital watermark; optimization model; data leakage traceability; data distribution

收稿日期：2022-03-28；**修订日期：**2022-11-11

引　言

数字指纹这一概念是由Wagner[1]在1983年发表的题为“Fingerprinting”的论文首次提出，在文献中介绍了数字指纹思想的相关术语以及分类，并且给出了一些数字指纹的例子，但其思想仍未能够在数字产品发生泄密时很好追踪到泄密者。1985年，由Blakley等[2]提出数据分发中出现的合谋攻击问题，攻击者通过对比发现数字产品中不同位置，并对其做出修改以此删除数字指纹。1998年，由Boneh等[3]在文献中针对合谋攻击提出了抗合谋编码(Anti-collusion codes，ACC)C安全码的概念，在不超过r($r\leqslant 3$)个非法用户合谋时，能够有效追踪到至少一个攻击者。同时，Trappe等[4]结合向量的正交化和组合理论提出了AND-ACC，将基于均衡不完全区组设计的BIBD(Balanced incomplete block design)码和正交码进行线性组合。由于BIBD的设计参数严于自由覆盖族码(Cover free family，CFF)区组，于是Li等[5]放宽了对区组设计的约束和参数设定，提出以BIBD的超集CFF码构造ACC码。Staddon等[6]还将C码与组合论相结合，提出了新的编码生成方法，能够追踪到多个攻击者，但其追踪的过程中误检率相对较高。Wang等[7]通过线性组合提出一种组内线性相关，组外线性无关的分组连续指纹编码。

数字指纹技术虽然得到了巨大的发展，但是依旧存在很多不足之处，很难满足实际应用需求。编码存在的主要不足有：(1)编码效率太低；(2)数字指纹存储空间太大；(3)泄密追踪复杂度过高；(4)抗合谋攻击能力太差。为了能够有效地追踪到泄密者，还需要进一步研究安全可行的数字指纹编码方案。第一个针对数值型数据的数据库水印算法由Agrawal等[8]在2002年首次提出，其主要通过修改数据的最低有效位(Least significant bit，LSB)来标记需要嵌入的水印位，算法对数据造成的的失真率较小。Zhang等[9]在2006年首次提出可逆的数据库水印的概念，是一种可以将嵌入水印的数据库进行还原的方案。张志彤等[10]提出了一种针对医学图像的基于奇偶性的大容量可逆水印算法。Alattar等[11]通过差分扩展可逆技术提出一种数据库可逆水印算法(Difference expansion watermarking，DEW)以此来增强水印的鲁棒性。2015年，Iftikhar等[12]提出一种鲁棒且半盲的可逆数据库水印技术(Robust and semi-blind reversible watermarking，RRW)，选取互信息最低的属性字段来嵌入水印，通过机器学习得到数据库中的最优水印信息。2015年，刘磊等[13]提出一种用于JPEG图像的高容量可逆水印算法，利用JPEG图像DCT系数的零系数间隔确定修改的位置。数据水印技术为数据安全提供了强有力的技术支持，在遭受到各种攻击[14]的情况下保证溯源的准确度都是研究的重点。2021年，宋岩等[15]提出一种新的可逆数据库水印方案，使用布谷鸟算法确定水印的嵌入位置，然后通过差分扩展技术实现嵌入操作。

数据库指纹嵌入技术虽然已经取得了长足的进展，但是依然存在着很多未解决的问题和不足之处，诸如嵌入容量小、失真度大、嵌入载体的范围小、抵抗合谋攻击能力弱等。在互联网、大数据和云存储飞速发展的背景下，数据库水印技术的发展也面临着巨大的挑战和机遇，仍需进一步研究与讨论。

数据的再分发指的是根据业务需求，已经嵌入数字指纹并分发给用户的数据副本需要再次分发的过程。数据的再分发过程需要对数据再次嵌入指纹编码，普通的指纹串联方法只能追踪到组内的合谋攻击，难以抵抗不同组内的合谋攻击，传统的抗合谋编码无法满足该应用场景，如图1所示。本文提出一种可再分发的抗合谋指纹编码方案，由于组内与组间合谋攻击的特点，组间使用BIBD编码，组内数据分发使用扩

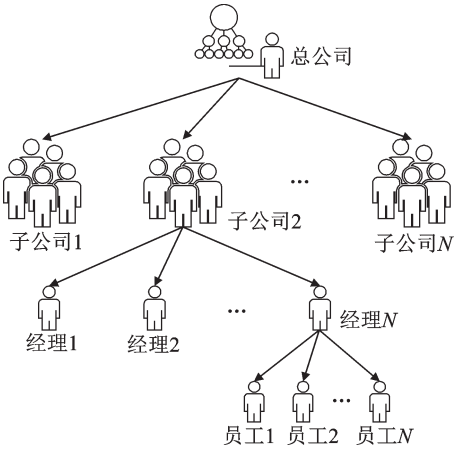

图1　数据分发过程

Fig.1　Data distribution process

展后的C码,构建可扩展的RD抗合谋编码。从而能够抵抗多个子组之间的合谋攻击、抵抗组内任意用户的合谋攻击,对泄密数据准确溯源。

1 可再分发抗合谋编码

基于可再分发抗合谋编码的低失真水印方案的具体框架如图2所示。

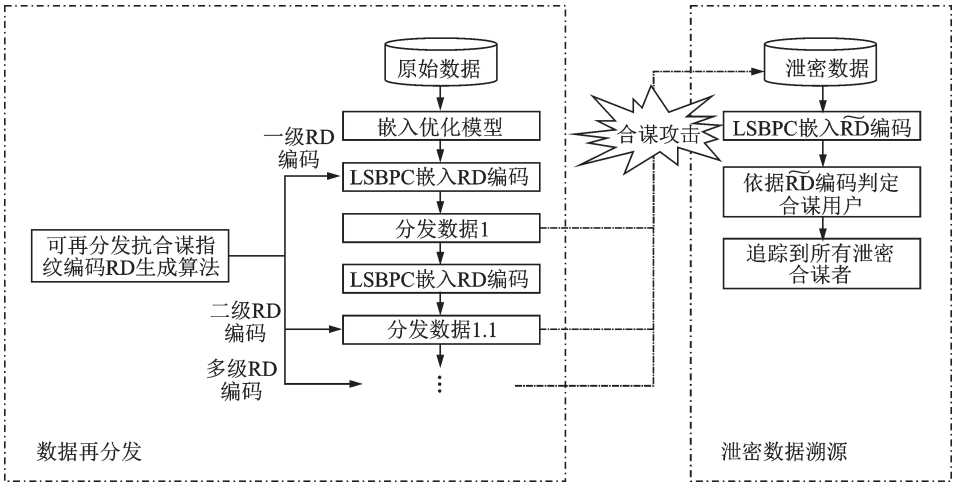

图2 方案框架图

Fig.2 Scheme frame diagram

本文提出可再分发抗合谋编码具体分发及嵌入步骤如下:

步骤1 优化分发嵌入结构。结合原始数据库与设置的RD编码参数生成优化模型,对优化模型进行求解,保存数据库与嵌入模型的对应关系。

步骤2 生成可再分发的抗合谋指纹编码。外码使用BIBD码,内码使用码字扩展的C编码,生成可再分发编码矩阵RD。

步骤3 低失真指纹嵌入。使用最低有效位奇偶修改算法LSBPC进行嵌入。

步骤4 数据泄密指纹提取。发现泄密数据,找到对应的嵌入密钥Key与嵌入结构矩阵$\boldsymbol{M}$,使用逆向LSBPC算法进行指纹提取,得到的指纹编码为$\widetilde{\mathrm{RD}}$。

步骤5 合谋用户泄密溯源。依据提取的抗合谋指纹编码$\widetilde{\mathrm{RD}}$,在对应的编码矩阵RD中溯源,依次得到泄密的分组,然后再溯源到组内的泄密用户。

1.1 可再分发抗合谋指纹编码方法

本文使用C码作为内码,追踪组内合谋泄密用户。C安全码是1995年提出的一种编码方案,全称为对数长度C安全码。码字设计为$\Gamma(n,d)$,参数n为码字的个数,d为最小汉明距离,码字的长度为$l=(n-1)d$。C安全码的设计如表1所示。虽然C码能够有效地抵抗“逻辑与”合谋攻击,但无法有效抵抗“逻辑或”合谋、平均合谋等合谋攻击方式。所以本节使用码字拓展技术对C码进行优化操作,将编码矩阵中所有的“1”和“0”分别替换为“10”和“01”。将原本逻辑“与”“或”操作后的码字状态“1”“1”“0”“0”,扩展为“10”“00”“11”“01”4种状态。本文以下说明使用外码BIBD(7,3,1),内码C码$\Gamma(3,1)$。

构建可再分发的抗合谋指纹RD编码,表示为RD(v,k,b,w),其中,v为外码BIBD编码长度,$k-1$为抗合谋人数,b为用户数,w为内码长度。可再分发抗合谋编码表示$\mathrm{RD}_{i,j}^{q}$,其表示第q级RD编码,第i

表1 $\Gamma(n,1)$C安全码

Table 1 $\Gamma(n,1)$ security code

UserID	b_1	b_2	…	b_{k-1}	b_k	…	b_{m-1}	b_m	…	b_{n-1}
u_1	1	1	…	1	1	…	1	1	…	1
u_2	0	1	…	1	1	…	1	1	…	1
…					…					
u_{k-1}	0	0	…	1	1	…	1	1	…	1
u_k	0	0	…	0	1	…	1	1	…	1
…					…					
u_{m-1}	0	0	…	0	0	…	1	1	…	1
u_m	0	0	…	0	0	…	0	1	…	1
…					…					
u_n	0	0	…	0	0	…	0	0	…	0

子组内的第j个编码。步骤如下：

(1) 构建第一级$\mathbf{RD}^1$、二级$\mathbf{RD}^2$抗合谋编码为

$$\mathbf{RD}^1=\mathrm{BIBD}=\begin{bmatrix}\mathrm{RD}_1^1\\ \mathrm{RD}_2^1\\ \mathrm{RD}_3^1\\ \mathrm{RD}_4^1\\ \mathrm{RD}_5^1\\ \mathrm{RD}_6^1\\ \mathrm{RD}_7^1\end{bmatrix},\ \mathbf{RD}^2=\begin{bmatrix}\mathrm{RD}_1^2\\ \mathrm{RD}_2^2\\ \mathrm{RD}_3^2\\ \mathrm{RD}_4^2\\ \mathrm{RD}_5^2\\ \mathrm{RD}_6^2\\ \mathrm{RD}_7^2\end{bmatrix}\begin{bmatrix}0&0&C&0&C&C&C\\ 0&C&0&C&C&0&C\\ 0&C&C&C&0&C&0\\ C&0&0&C&0&C&C\\ C&0&C&C&C&0&0\\ C&C&0&0&C&C&0\\ C&C&C&0&0&0&C\end{bmatrix}$$

(2) 构建多级再分发$\mathbf{RD}^q$抗合谋编码为

$$\mathbf{RD}^q=\left[\mathbf{RD}^1\right]\cdots\begin{bmatrix}0&0&\mathrm{RD}^{q-1}&0&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}\\ 0&\mathrm{RD}^{q-1}&0&\mathrm{RD}^{q-1}&\mathrm{RD}^{k-1}&0&\mathrm{RD}^{q-1}\\ 0&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&0&\mathrm{RD}^{q-1}&0\\ \mathrm{RD}^{q-1}&0&0&\mathrm{RD}^{q-1}&0&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}\\ \mathrm{RD}^{q-1}&0&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&0&0\\ \mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&0&0&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&0\\ \mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&\mathrm{RD}^{q-1}&0&0&0&\mathrm{RD}^{q-1}\end{bmatrix}\tag{1}$$

例如，公司内部使用多级抗合谋编码来处理数据的分发操作，公司C_1需要将数据分发给下属员工使用，所以嵌入了两级RD编码；公司C_2需要高质量的数据，所以获取的数据只嵌入了一级RD编码，后期为了业务要求需要将数据再分发给下属员工使用，则公司C_2通过授权获取嵌入两级RD编码的数据，再分发给员工。基于BIBD与C码的可再分发的混合编码RD编码，实现按需获取只嵌入一级编码的低失真数据，或嵌入二级编码的二次分发数据。将BIBD码长短、能够抵抗用户与、或合谋的优点与C码构造简单能够抵抗任意用户与合谋的优点相结合。生成的抗合谋指纹RD编码能够适应公司内部组间合谋攻击概率低、组内合谋概率高的特点，如小组无需再分发，可按需获得失真率低的数据。

1.2 基于RD编码的合谋者判定

一旦嵌入RD编码的数据发生泄密，可提取其中的编码$\widetilde{\mathrm{RD}}$进行溯源，追踪到所有参与泄密的人。如表2的两级RD编码的合谋判定主要分为两步进行：(1)一级编码。首先由一级RD编码RD_1外码确定合谋的子组，例如，泄密数据中提取到的残留一级编码为$\widetilde{\mathrm{RD}}_0=\mathrm{RD}_1^1\cap\mathrm{RD}_2^1$，以此追踪到泄密子组1

和2;(2)二级编码。由一级编码可知两个子组的编码$RD_1^1=(0010111)$,$RD_2^1=(0101101)$,如果$rd_i=1(rd_i\in RD_1^1\oplus RD_1^1)$,$RD_1^1\oplus RD_2^1=(0111010)$,则由此位置确定子组内的合谋情况。例如$i=2$,$RD_1^1$码比特为0并且$RD_2^1$码比特为1,可通过内码判断出子组2内的合谋情况,此位置的合谋内码为$\tilde{C}=C_2\cap C_3$,从而追踪到子组2内的合谋人。再如$i=3$,RD_1^1码比特为1并且RD_2^1码比特为0,可通过内码判断出子组1内的合谋情况,此位置的合谋内码为$\tilde{C}=C_1$,从而追踪到子组1内的合谋者。

表2 两级RD抗合谋编码

Table 2 Two-level RD anti-collusion coding

一级编码			二级编码							
第一组编码	RD_1^1	0010111	C内码		0	0	C	0	C	C	C
			C_1	000	000	000	000	000	000	000	000
			C_2	100	000	000	100	000	100	100	100
			C_3	110	000	000	110	000	110	110	110
第二组编码	RD_2^1	0101101	C内码		0	C	0	C	C	0	C
			C_1	000	000	000	000	000	000	000	000
			C_2	100	000	100	000	100	100	000	100
			C_3	110	000	110	000	110	110	000	110

数据再分发之后,发生合谋可能存在子组内,也有可能发生在不同子组之间,接下来针对两级RD抗合谋指纹编码,讨论在不同情况下的合谋检测情况。为了更清楚展示合谋检测的过程,对表2中任意两组的RD抗合谋编码进行合谋攻击,并分析同组合谋攻击、不同组合谋攻击结果。第一组的外码为010111,将1的位置替换为内码形成二级编码00C0CCC,因为内码C采用的是3*3的码矩阵,最后得到二级编码矩阵为表2中第一组有底色部分。以下举例说明两种合谋攻击的判定。

(1) 组内合谋攻击

组内的合谋情况不会导致一级抗合谋编码的改变,所以提取一级抗合谋的改变与否,可直接判断组内组间的合谋情况,如表3所示,其中灰色背景部分展示内码的逻辑合谋。例如以下情况分配给组1内用户的编码为$U_{1,1}=\{RD_1^1,\ RD_1^2\}$,$U_{1,2}=\{RD_1^1,\ RD_2^2\}$,$U_{1,3}=\{RD_1^1,\ RD_3^2\}$,组内发生用户合谋攻击,不会改变能标识分组的编码$RD_1^1$。反之,如果从泄密数据中标识分组的编码$RD_1^1$不发生改变,能够溯源到组别1,则发生泄密的情况在组1内部,组间未发生合谋。组内发生合谋泄密,提取的组内合谋编码为$\widetilde{RD^2}$与$RD_1^2\cap RD_2^2\cap RD_3^2$合谋集相同,即可追踪到合谋泄密的用户$U_{1,1}$、$U_{1,2}$、$U_{1,3}$。

(2) 组间合谋攻击

组间的合谋情况比较复杂,包括组与组之间的合谋、不同组内用户的合谋,最大的可检测合谋组数为$k-1$,具体合谋情况如表4所示。首先依据一级编码判断合谋情况,依据$RD_1^1\cap RD_2^1$可知是第1组和第2组之间发生合谋。例如以下情况分配给用户的编码为$U_{1,1}=\{RD_1^1,\ RD_1^2\}$,$U_{1,2}=\{RD_1^1,\ RD_2^2\}$,$U_{2,1}=\{RD_2^1,\ RD_1^2\}$,组间发生用户合谋的编码,将会改变能标识分组的编码RD_1^1为$RD_1^1\cap RD_2^1$,一级编码的合谋集是唯一的,从而可以溯源到组别1和2。组别1和2的标识编码为$RD_1^1=\{0010111\}$,$RD_2^1=\{0101101\}$,溯源到合谋组别之后,再对合谋组1内进行溯源,取RD_1^1码比特位置为1且RD_2^1码比特位置为0的二级编码位置的内码$\tilde{C}$,此时与合谋集$C_1\cap C_2$相同,从而溯源到1组内用户$U_{1,1}$和$U_{1,2}$。同理,对组2进行组内合谋泄密溯源,得到泄密用户$U_{2,1}$。最终,组间合谋用户包括$U_{1,1}$、$U_{1,2}$和$U_{2,1}$。

表3 组内合谋攻击实例分析

Table 3 Analysis of the case of collusion attack in the group

攻击方式	二级编码						
$U_{1,1}\cap U_{1,2}\cap U_{1,3}$	0	C_1	$C_1\cap C_2\cap C_3$	0	$C_1\cap C_2\cap C_3$	$C_1\cap C_2\cap C_3$	$C_1\cap C_2\cap C_3$
	010101	010101	111101	010101	111101	111101	111101
$U_{1,1}\cap U_{1,2}$	0	0	$C_1\cap C_2$	0	$C_1\cap C_2$	$C_1\cap C_2$	$C_1\cap C_2$
	010101	010101	110101	010101	110101	110101	110101
$U_{1,1}\cup U_{1,2}\cup U_{1,3}$	0	0	$C_1\cup C_2\cup C_3$	0	$C_1\cup C_2\cup C_3$	$C_1\cup C_2\cup C_3$	$C_1\cup C_2\cup C_3$
	010101	010101	111101	010101	111101	111101	111101
$U_{1,1}\cup U_{1,2}$	0	0	$C_1\cup C_2$	0	$C_1\cup C_2$	$C_1\cup C_2$	$C_1\cup C_2$
	010101	010101	110101	010101	110101	110101	110101

表4 组间合谋攻击实例分析

Table 4 Analysis of the case of collusion attack between groups

攻击方式	二级编码						
$U_{1,1}\cap U_{1,2}\cap U_{2,1}$	0	C_1	$C_1\cap C_2$	C_1	$C_1\cap C_2$	$C_1\cap C_2$	$C_1\cap C_2$
	010101	010101	110101	010101	110101	110101	110101
$U_{1,1}\cap U_{1,2}\cap U_{2,3}$	0	$0\cap C_3$	$C_1\cap C_2$	$0\cap C_3$	$C_1\cap C_2\cap C_3$	$C_1\cap C_2$	$C_1\cap C_2\cap C_3$
	010101	111101	110101	111101	111101	110101	111101
$U_{1,1}\cup U_{1,2}\cup U_{2,1}$	0	C_1	$C_1\cup C_2$	C_1	$C_1\cup C_2$	$C_1\cup C_2$	$C_1\cup C_2$
	010101	010101	110101	010101	110101	110101	110101
$U_{1,1}\cup U_{1,2}\cup U_{2,3}$	0	C_3	$C_1\cup C_2$	C_3	$C_1\cup C_2\cup C_3$	$C_1\cup C_2$	$C_1\cup C_2\cup C_3$
	010101	111101	110101	111101	111101	110101	111101

2 一种低失真的数据库指纹嵌入与提取算法

对于第1节提出的多级抗合谋编码，每一级编码都有不同的嵌入位置s_i、嵌入的阈值ε_i；同时随机密钥可能带来分组不均等问题，导致数据库水印方法鲁棒性差的问题。本节针对可再分发抗合谋编码提出一种嵌入优化模型，优化嵌入结构，使用低失真嵌入方法嵌入水印，即使少量数据也能保证水印算法的鲁棒性。

设一个数据库关系为$R(P, A_1, A_2, \cdots, A_j, \cdots, A_v)$，其中$P$为主键，$A_j$为数据库$R$的属性，$r_i$为数据库$R$中元组，$r_iA_j$为元组$r_i$的第$j$个属性值$A_j$。设属性值$r_iA_j$在满足嵌入精度$\varepsilon$情况下，修改LSB位，不会影响数据库的正常使用。设待嵌入的数据库水印为一个L长度的二进制串：$W=\{w_0, w_1, \cdots, w_l, \cdots, w_0\}$ $(0\leqslant l\leqslant L-1)$。使用hash函数为hash()，密钥为Key，满足嵌入冗余位的分组编号为id_i。

2.1 数据库嵌入优化模型

目前衡量指纹嵌入算法的性能指标有：鲁棒性、嵌入容量大小、数据失真大小，对数据库嵌入算法建立优化模型，分别使用3个目标函数来衡量上述指标。目标函数f_1期望最优密钥将哈希分组尽可能均匀，提高嵌入算法的鲁棒性；目标函数f_2将嵌入容量与提出的抗合谋编码分发结构相结合考虑，保证在分发结构下嵌入容量最大；目标函数f_3希望嵌入后的数据与原始数据的改变量最小。结合以上3个嵌入规则，3个目标函数之间存在矛盾的关系，在密钥最优的情况下，保证嵌入容量尽可能大、数据失真尽可能小，建立数字指纹嵌入多目标优化模型，具体公式为

$$\begin{cases} f_1 = \mathrm{std}\left(\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}\right) \\ f_2 = \mathrm{std}\left(\dfrac{\sum\limits_{i}^{\varepsilon_1}\sum\limits_{j}^{n} M_{i,j}^{1}}{L(\mathrm{RD}^1)}, \dfrac{\sum\limits_{i}^{\varepsilon_2}\sum\limits_{j}^{n} M_{i,j}^{2}}{L(\mathrm{RD}^2)}, \cdots, \dfrac{\sum\limits_{i}^{\varepsilon_n}\sum\limits_{j}^{n} M_{i,j}^{n}}{L(\mathrm{RD}^n)}\right) \\ f_3 = \sum\limits_{i}^{\varepsilon_1}\sum\limits_{j}^{n} M_{i,j}^{1}\cdot i + \sum\limits_{i}^{\varepsilon_2}\sum\limits_{j}^{n} M_{i,j}^{2}\cdot i \cdots + \sum\limits_{i}^{\varepsilon_n}\sum\limits_{j}^{n} M_{i,j}^{n}\cdot i \\ \quad \text{满足:} \min_{\varepsilon} \leqslant \varepsilon_{i(i=1,\cdots,n)} \leqslant \max_{\varepsilon};\ \mathrm{key}\in\forall;\ 0 \leqslant s_{i(i=1,\cdots,n)} \leqslant 2^n \end{cases} \tag{2}$$

式中:c_i为第i个哈希分组种可嵌入码比特的个数,std()为分组个数的方差,$\boldsymbol{M}$为整个数据可嵌入位置矩阵,$\boldsymbol{M}_{i,j}^{1}$为嵌入结构s_1所对应的可嵌入位置子矩阵;$L(\mathrm{RD}^n)$为抗合谋编码的长度;ε_i为对应的i层编码嵌入数据失真的阈值,key为分组密钥。

2.2 数字指纹嵌入提取算法

本节提出一种最低有效位奇偶改变(Least significant bit parity change, LSBPC)数字指纹算法,首先提取数据库中所有满足嵌入要求的位置,然后对其进行哈希分组,按照一定的嵌入密度对嵌入位置的最低有效位(Least significant bit, LSB)、次低有效位(Sub-low significant bit, SLSB)根据嵌入bit的不同改变其LSB的奇偶性,嵌入步骤如算法1所示。

数字指纹的嵌入实例如图2左侧所示,嵌入比特"1"需要将对应数值的最低有效位和次低有效位的奇偶性修改为不同,嵌入比特"0"需要将对应数值的最低有效位和次低有效位的奇偶性修改为相同,根据概率估计只需要修改一半嵌入位置的最低有效位就可以嵌入所有的数字指纹。数据库水印的提取算法是嵌入算法的逆过程,提取数据库中嵌入位置,并进行哈希分组,同时对同一分组中的奇偶性进行大数表决,若该组奇数多则提取水印位为1,否则为0,数字指纹提取算法步骤如算法2所示。

如图3所示是一个基于最低有效位奇偶改变的数字指纹算法的简单实例。原始的数据库中有3个字段A_j(属性1,属性2,属性3)和4个元组r_i。数字指纹为f=10011长度为5,则哈希分组的数目也是5,嵌入阈值ε为3(即小数点后3位)。则首先遍历所有的属性值判断是否满足嵌入阈值,满足的数据计算其分组编号group = hash(Key$\|r_iP\|r_iA_j$)。然后,依据对应的指纹位置码字,修改属性值r_iA_j的最低有效位奇偶性,达到嵌入指纹码字的目的,最后得到嵌入指纹后的分发数据DB_{EF}。例如,0.455被哈希分组到1组,需要嵌入码字"1",因为LSB为5,SLSB为5,两者奇偶性需要不同,则将SLSB − 1。同样,指纹提取的时候是一个逆过程,按照嵌入算法同样的分组方法得到分组号group,不同的是,需要判断最低有效位的奇偶性,如果奇偶性相同,则当前属性值提取到"0"码字,否则当前属性值提取到"1"码字,将提取的码字追加到对应的分组,最后f_{group}为group分组中出现最多的码字。

算法1 指纹嵌入算法

输入:原始数据库DB,待嵌入指纹f,阈值ε,分组密钥key。

输出:嵌入指纹的分发数据库DB_{EF}。

(1) DB_{EF} = [ ]; //初始化嵌入指纹后的分发数据库

(2) For each r_iA_j in R LSBPC, //遍历每一个属性值

(3) If r_iA_j not satisfy redundancy ε do //属性值不满足嵌入条件

(4) Continue //则继续遍历

(5) group = hash(Key$\|r_iP\|r_iA_j$) //对属性值哈希分组

(6) End if

(7) If $f_{\text{group}} = 1$ do //对应的指纹bit为1
(8) //最低有效与次低有效位奇偶性不同
(9) If even or old(SLSB, SLB)do
(10) Continue
(11) Else
(12) $r_i A_j = r_i A_j + 1$ //修改属性值的最低有效与次低有效位奇偶性为不同
(13) End if
(14) End if
(15) If $f_{\text{group}} = 0$ do //如果分组group对应的指纹bit为0
(16) If even or old(SLSB, SLB)do //奇偶性不同
(17) $r_i A_j = r_i A_j - 1$ //修改最低有效奇偶性
(18) Else
(19) Continue
(20) End if
(21) End if
(22) DB_{EF}.append(r_i)
(23) End for
(24) Return DB_{EF}

算法2 数字指纹提取算法

输入:泄密数据库DB,阈值ε,分组密钥key。
输出:提取指纹f。

(1) $f = []$, $w = []$; //提取指纹f,初始化二维数组w存分组中提取的码字
(2) $\text{count}_0 = 0$, $\text{count}_1 = 1$; //Even counter
(3) For each $r_i A_j$ in R do// //遍历每一个属性值
(4) If $r_i A_j$ satisfy redundancy ε do //属性值满足嵌入阈值
(5) group = hash(Key‖$r_i P$‖$r_i A_j$) //对属性值哈希分组
(6) If even or old(SLSB, SLB)do //奇偶性不同
(7) w_{group}.append(1) //group组中提取到一个1bit码字
(8) Else
(9) w_{group}.append(0) //group组中提取到一个0bit码字
(10) End if
(11) End if
(12) End for
(13) For each group in w do // 遍历中间码字二维数组
(14) For each g_i in group do // 遍历每组中的所有码字
(15) $\text{count}_0 \leftarrow \text{count}_0 + 1$ 如果g_i是偶数
(16) $\text{count}_1 \leftarrow \text{count}_1 + 1$ 如果g_i是奇数
(17) If $\text{count}_0 \geqslant \text{count}_1$ do //大数判决f_i的码比特值
(18) $f_i = 0$

(19) Else
(20) $f_i = 1$
(21) End if
(22) End for
(23) End for
(24) Return f

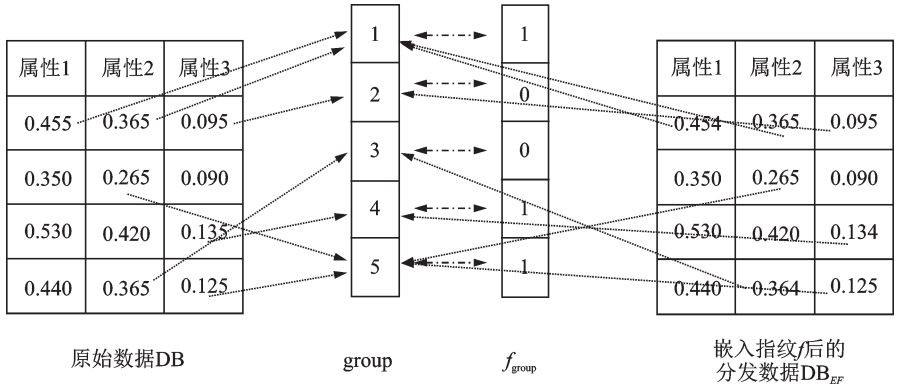

图3 LSBPC指纹算法示意图

Fig.3 Schematic diagram of LSBPC fingerprint algorithm

3 实验与分析

实验设计主要验证以下几个问题:

(1)数字指纹嵌入优化模型会带来多大的优势,LSBPC算法会对数据造成多大的失真;

(2)可再分发抗合谋指纹编码在常见合谋攻击后,溯源准确率与误判率;

(3)与同类方法实验对比如何。

本节将公开数据集Abalone Data Set作为分发数据的源数据,对该数据集建立三目标嵌入优化模型,并通过LSBPC的方式进行嵌入数字指纹编码,比较当前的嵌入算法与其他算法的失真率,分析其鲁棒性和数据的可用性;对比双层Tardos[16]编码和CCF-Ⅰ级联编码[17]在相同嵌入率的情况下,不同载体攻击下的数据泄密溯源准确率与误判率;使用不同的RD编码,分析其在合谋攻击后的溯源准确率。

3.1 编码性能分析

码字扩展后的Ⅰ编码的码长为w*2;允许分发人数为w;允许最大合谋人数为w;能够追踪到任意用户之间的合谋,但是码长与用户呈线性增长的关系。

为降低检测所需的计算成本,引入了双层Tardos编码结构,$M_1 \times M_2$个用户被分为M_1个区组,每个区组包含M_2个用户。利用双层的编码特点有效缩小了指控合谋者的搜索域,性能评价和分析表明方案在减小计算成本和检测性能上具有优越性。指纹长度$m = m_1 + m_2$,最大合谋人数与组内编码相关,该方法有编码短构造方法简单、码长短等优点,但是缺点也很明显,对于不同组但是组内位置相同的用户合谋无法检测,例如第1区组的第1个用户与第2区组的第1个用户合谋攻击是无法检测的。

将CFF码作为内码、Ⅰ码作为外码进行级联编码得到CFF-Ⅰ编码,提高了编码效率,能够抗击r个用户合谋攻击、同时能抵抗与合谋、或合谋等多种合谋攻击。

可再分发的抗合谋编码RD-ACC表示为RD(v, k, b, w)。一级编码为外码BIBD编码,码长达到最优v,最大抗合谋人数为$k-1$,分发人数为b;二级编码为级联编码,码长为(w*2)*v,允许分发人数为

w*v,最大抗合谋人数为($k-1$)*w;多级RD编码的码长为(w*2)$^{n-1}$*v,允许分发人数为w^{n-1}*v,最大抗合谋人数为($k-1$)*w^{n-1}。

3.2 嵌入优化模型实验

在该数据集下,给出可嵌入结构矩阵$\boldsymbol{M}$,其中$\boldsymbol{M}_{ij}$表示第j个字段满足阈值$\varepsilon_i=\{4,3,2,1\}$的可嵌入位置个数。

$$\boldsymbol{M}=\begin{bmatrix}0&0&0&0&243&234&255&17&0\\0&227&256&248&230&230&223&229&0\\0&243&209&230&23&34&20&225&0\\0&31&35&22&4&2&2&29&0\end{bmatrix},\ \boldsymbol{S}=\begin{bmatrix}0&1&1&2&1&2&1&1&0\\0&2&2&2&2&2&2&2&0\\0&0&0&2&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{bmatrix}$$

对Abalone建立优化嵌入模型,通过遗传算法求解最优解:最优密钥为“152840287.50796878”,嵌入结构选择为S,对应的3个目标函数值为[17.919 5, 1.678 5, 4 725]。

本节对原始数据采用不同的数字指纹嵌入算法,嵌入相同的指纹10次,对比数据的平均失真率。如表5所示,LSB嵌入方法是一种低失真嵌入方法,本身嵌入的失真已经比较小,对比LSBPC嵌入方法的失真率。LSB方法是修改最低有效位,依据嵌入的比特,最低有效位的修改范围(0~9)之间。LSBPC最低有效位奇偶性修改方法,只修改最低有效位+1或-1操作,修改范围更小。所以LSBPC数据均值失真率大概是LSB的10%,同时,方差也更小。

表5 抗合谋指纹嵌入低失真实验

Table 5 Anti-collusion fingerprint embedding low distortion experiment

字段	均值改变量/%		标准差改变量/%	
对比项	LSB方案	LSBPC方案	LSB方案	LSBPC方案
Length	3.170 900E-02	1.375 705E-03	-2.028 362E-01	-1.211 088E-03
Diameter	-1.198 560E-07	-3.097 777E-07	-1.346 830E-07	-2.269 559E-07
Height	-2.362 880E-07	1.326 860E-07	9.389 690E-07	2.448 710E-07
Whole weight	-1.261 480E-04	5.119 003E-06	9.061 310E-06	6.290 000E-07
Shucked weight	-2.264 060E-04	-1.385 470E-05	-4.219 870E-06	1.586 058E-06
Viscera weight	-5.745 050E-04	-1.548 747E-06	1.101 500E-04	2.787 484E-06
Shell weight	-1.136 660E-04	-2.175 262E-05	-3.701 250E-05	1.902 301E-07
Rings	4.701 990E-09	0.000 000E+00	-2.112 470E-09	0.000 000E+00

如图4所示,对本文中提出的水印嵌入优化模型进行实验,对比优化后的密钥和随机密钥对数据分组的影响。随机密钥产生的分组中数据个数在20到103个之间浮动,但是,优化密钥产生的随机分组中数据的个数在38到82之间浮动。实验结果很明显:随机密钥数据分组后,每组数据的个数随机性很大,但是影响水印鲁棒性的往往是数据分组个数较小的一组,类似于“水桶的短板效应”。相较于随机密钥,优化后的密钥分组更为均匀,在数据库遭受到攻击后也能保持较好的鲁棒性。

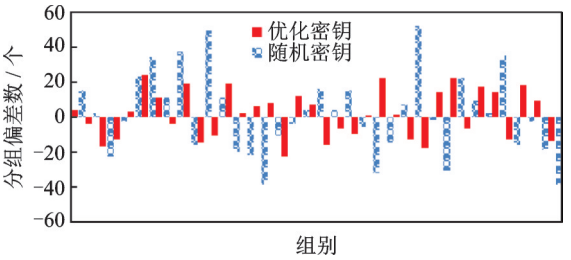

图4 随机密钥与优化密钥对分组的影响

Fig.4 Effect of random key and optimized key on grouping

为了验证不同密钥对数字指纹鲁棒性的影响,设置不同数据量(100%、20%和10%的Abalone Data Set)的数据集作为载体,在随机密钥和最优密钥两种分组下,嵌入两级可再分发抗合谋

编码,对载体数据进行不同强度的载体攻击。分别对嵌入数字指纹的数据进行3种常见的载体攻击(添加、删除、修改攻击),在不同的载体攻击强度下比较提取的码字准确率。

如图5所示,在不经过优化模型下,对不同量级(100%,20%,10%)的数据,分别实验了不同强度的3种载体攻击后编码提取的准确度。在数量级为100%的数据集作为载体的情况下,即使是3种载体攻击强度达到90%,码字提取的准确率也能保证在90%左右。在数量级是20%的作为载体的情况下,对载体进行不同强度的攻击。此时,元组攻击后的码字准确率率先在攻击强度为40%时就出现较大波动,元组删除攻击后的码字准确率也在攻击强度为60%时出现迅速降低。在数量级为10%时,在攻击强度超过25%之后码字提取的准确率就迅速降低。可发现,随着数据集数量级的降低,在20%数据集下,码字准确度具有较大的随机性。这主要是因为随机密钥导致分布不均,与水桶的"短板效应"类似,水印算法的鲁棒性往往与分组中元素最少的相关。

如图6所示,经过优化后的水印算法,通过对比图5可以明显发现,在数量级为100%的数据集作为载体收到攻击后码字提取的准确率与随即密钥类似,攻击强度达到90%依然有很高的准确率。当数量级变小,在10%、20%数据集的情况下,即使攻击强度达到60%,也能基本保证码字的准确性。在攻击强度增大的情况下,码字提取准确性较为平缓下降,不会出现较大"波动"情况。综合对比,优化模型能很大程度保证算法在分组时,每组数据的个数大致相同,在常见的载体攻击后也能保证码字提取的准确度。

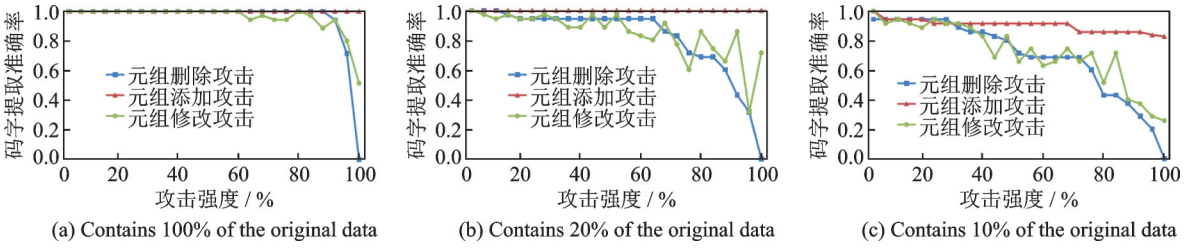

图5 未进行优化嵌入数字指纹情况下载体攻击后码字提取准确率

Fig.5 Codeword extraction accuracy after carrier attack without optimizing the embedded digital fingerprint

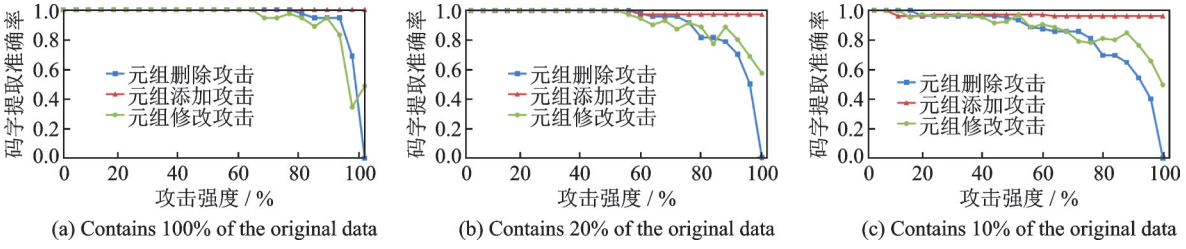

图6 进行优化后嵌入数字指纹情况下载体攻击后码字提取准确率

Fig.6 Codeword extraction accuracy after carrier attack with optimizing the embedded digital fingerprint

3.3 合谋检测准确率及误判率

检测率与误判率是衡量指纹检测性能的重要指标,分别用p_{d}、p_{fp}表示。实验对比不同抗合谋编码在不同合谋人数的随机合谋攻击情况下,对比合谋人的检测率和误判率,分别如图7、8所示。

对比3种不同级联编码正确检测率与错误检测率。码字扩展后的CCF-Ⅰ:4-CFF(16,20)码和10*10大小的Ⅰ码作为内码,码长320 bit,用户数为200,理论最大抗最大合谋人数30(需要是不同组的);双层Tardos编码:采用文献[16]中服从正态分布的码长为5 000,用户数为100的正交双层Tardos编码,理论最大抗最大合谋人数30(需要是不同组的);本文提出的RD编码:4-BIBD(16,20)外码和10*10大小的C码作为外码,一级编码的码长16 bit,二级编码的码长320 bit。实验结果表明,CCF-Ⅰ和双层Tardos编码

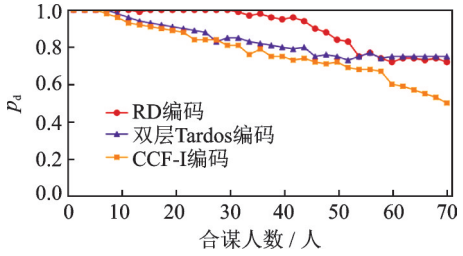

图 7 不同编码方案合谋者检测率对比

Fig.7 Comparison of detection rates of conspirators in different coding schemes

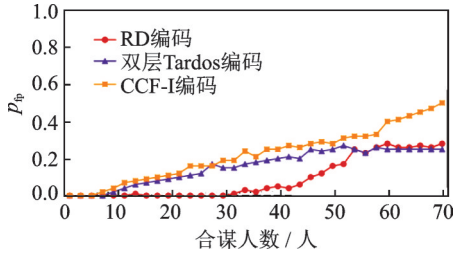

图 8 不同编码方案合谋者误判率对比

Fig.8 Comparison of misjudgment rates of colluders in different coding schemes

在任意人数合谋的情况下，合谋人数超过 10 人之后，正确检测率 p_d 就开始下降，错误检测率 p_{fp} 开始上升。这是因为对于这两种级联编码无法识别不同子组中相同位置编码合谋情况，导致出现误检的情况。

对比不同 RD 编码正确检测率 p_d 与误判率 p_{fd}。RD(16,4,20,10)：4-BIBD(16,20)外码和 10*10 大小的 C 码作为外码，一级编码的码长 16，二级编码的码长 320，理论最大抗合谋人数是 30。RD(46,6,69,10)：6-BIBD(46,69)外码和 10*10 大小的 C 码作为外码，一级编码的码长 46；二级编码的码长 920，理论最大抗合谋人数是 50。RD(49,7,56,10)：7-BIBD(49,56)外码和 10*10 大小的 C 码作为外码，一级编码的码长 49，二级编码的码长 980，理论最大抗合谋人数是 60。

如图 9 和图 10 所示，比较不同外码构造的 RD 编码的合谋检测正确率 p_d 和误判率 p_{fp}，比较 3 种 RD 编码：RD(16,4,20,10)、RD(46,6,69,10)、RD(49,7,56,10)。内码确定的情况下，最大可抗合谋用户数目与外码正相关，外码能够抵抗任意子组之间的合谋，内码能抵抗组内任意个泄密人的合谋攻击。一级 RD 结合二级 RD 编码，在增加部分码长的情况下，能够抵抗任意子组内的任意泄密人的合谋攻击。

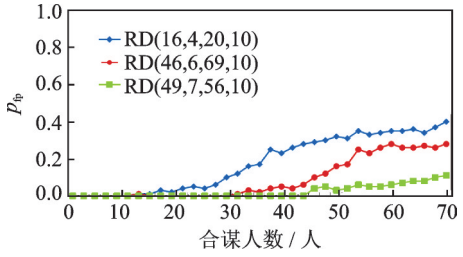

图 9 不同 RD 编码的合谋人数检测率

Fig.9 Detection rate of colluding people with different RD codes

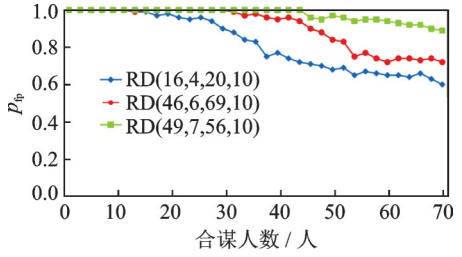

图 10 不同 RD 编码的合谋人数误判率

Fig.10 Misjudgment rate of colluding people with different RD codes

4 结束语

为了满足现实中用户因为数据失真，再分发数据的需求，本文提出一种基于可再分发抗合谋指纹编码的数据泄密溯源方案。综合现实场景中数据再次分发(公司-小组-组员多级分发场景)的切实需求，考虑小组之间联系弱、发生合谋攻击可能性较小，同组组员之间的联系强、发生合谋概率可能性很大的特点。本文使用 BIBD 作为外码，码字扩展后的 C 码作为内码，构建一种可再分发的抗合谋指纹编码 RD-ACC。在此基础上，提出一种基于多目标优化的数据库指纹算法，算法优化数据失真、嵌入结构和鲁棒性这 3 个目标。能够在较小的数据库失真情况下保证数字指纹较高的鲁棒性，提取出的 RD-ACC 能够有效抵抗组内、组间多用户合谋攻击。

参考文献:

[1] WAGNER N R. Fingerprinting[J]. IEEE Symp Security & Privacy, 1983, 1: 18-22.

[2] BLAKLEY G R, MEADOWS C, PURDY G B. Fingerprinting Long forgiving messages[C]//Proceedings of Advances in Cryptology-CRYPTO 85. California:[s.n.], 1985: 180-189.

[3] BONEH D, SHAW J . Collusion-secure fingerprinting for digital data[J]. IEEE Transactions on Information Theory, 1998, 44 (5): 1897-1905.

[4] TRAPPE W, WU M , WANG Z L, et al. Anti-collusion fingerprinting for multimedia[J]. IEEE Transactions on Signal Processing, 2003, 51(4): 1069-1087.

[5] LI Q, WANG X, LI Y, et al. Construction of anti-collusion codes based on cover-free families[C]//Proceedings of the Sixth International Conference on Information Technology. Las Vegas: IEEE, 2009.

[6] STADDON J N, STINSON D R, WEI R. Combinatorial properties of frameproof and traceability codes[J]. Information Theory IEEE Transactions on, 2001, 47(3): 1042-1049.

[7] WANG Z J, TRAPPE W, MIN W, et al. Group-oriented fingerprinting for multimedia forensics[J]. Eurasip Journal on Advances in Signal Processing, 2004, 2004(14): 1-21.

[8] AGRAWAL R , KIERNAN J. Watermarking relational databases[C]//Proceedings of the 28th International Conference on Very Large Databases.[S.l.]:[s.n.],2002: 155-166.

[9] ZHANG Y, YANG B, NIU X M . Reversible watermarking for relational database authentication[J]. Journal of Computers, 2006, 17(2): 59-65.

[10] 张志彤, 赵耀, 倪蓉蓉,等 . 奇偶性分类下大容量医学图像可逆水印算法[J]. 数据采集与处理, 2013, 28(5): 626-632.
ZHANG Zhitong, ZHAO Yao, NI Rongrong, et al. A reversible watermarking algorithm for high-capacity medical images based on the parity of data[J]. Data Acquisition and Processing, 2013, 28 (5): 626-632.

[11] ALATTAR A M . Reversible watermark using difference expansion of quads[C]//Proceedings of Acoustics, Speech, and Signal Processing.[S.l.]:IEEE ,2004.

[12] IFTIKHAR S, KAMRAN M. RRW-A robust and reversible watermarking technique for relational data[J]. IEEE Transactions on Knowledge & Data Engineering, 2015, 27(4): 1-14.

[13] 刘磊, 赵耀, 倪蓉蓉,等 . 一种用于JPEG图像的高容量可逆水印算法[J]. 数据采集与处理, 2015, 30(4): 816-823.
LIU Lei, ZHAO Yao, NI Rongrong, et al. High-capacity reversible watermarking scheme for JPEG images[J]. Data Acquisition and Processing, 2015, 30(4): 816-823.

[14] 杨蕾 . 基于数据库水印的数据溯源技术研究[D]. 天津:天津理工大学, 2019.
YANG Lei. Research on data traceability technology based on database watermarking[D]. Tianjin: Tianjin University of Technology, 2019.

[15] 宋岩,沈泉江,杨洪山 . 基于布谷鸟算法的可逆数据库水印方案[J]. 计算机应用与软件,2021,38(12): 304-313.
SONG Yan, SHEN Quanjiang, YANG Hongshan. A reversible database watermarking scheme based on the cuckoo algorithm [J]. Computer Applications and Software, 2021, 38(12): 304-313.

[16] 张恒 . 抗合谋数字指纹编码与检测研究[D]. 长沙:湖南大学,2011.
ZHANG Heng. Research on anti-collusion digital fingerprint coding and detection[D]. Changsua:Hunan University, 2011.

[17] 李启南, 董一君, 李娇,等 . 基于CFF码和I码的抗合谋数字指纹编码[J]. 计算机工程, 2015(6): 110-115.
LI Qinan, DONG Yijun, LI Jiao, et al. Anti collusion digital fingerprint coding based on CFF code and I code[J]. Computer Engineering, 2015(6): 110-115.

作者简介:

邸云龙(1996-),男,硕士研究生,研究方向:数据与内容安全,E-mali:2544692079@qq.com。

张迎周(1978-),**通信作者**,男,博士,教授,硕士生导师,研究方向:软件与信息安全,E-mail: zhangyz@njupt.edu.cn。

汪天琦(1997-),女,硕士研究生,研究方向:数据与内容安全。

李鼎文(1998-),男,硕士研究生,研究方向;数据与内容安全。

朱林林(1997-),女,硕士研究生,研究方向:数据与内容安全。

(编辑:夏道家)