有限域上高效的细粒度数据完整性检验方法

陈 龙[1] 王国胤[2]

[1](重庆邮电大学计算机取证研究所 重庆 400065)

[2](重庆邮电大学计算机科学与技术研究所 重庆 400065)

摘 要 基于交叉检验思想的细粒度数据完整性检验方法在实现完整性检验的同时可以对少数错误进行准确和高效的隔离,从而避免因偶然错误或个别篡改造成整体数据失效的灾难性后果.针对需要隔离多个错误时现有方案效率不高的问题,提出了多维结构下基于有限域均匀划分的完整性交叉检验方法,相应地构造了高效的多错完整性指示编码.该方法将完整性检验 Hash 数据分为若干组,任一组 Hash 可在某一中间粒度独立指示所有数据对象的完整性,多组 Hash 结合起来则在更小的基本粒度指示数据的完整性.该方法实现了模块化的 Hash 结构,对于 $GF(q)$ 上的 d 维向量空间,每增加 $(d-1)$ 组共 $(d-1)q$ 个 Hash 即可多指示一个错.分析了该编码在不同参数下的性能,分析结论和实验结果表明该编码效率高,具有灵活的参数选择,可满足各种应用的不同需要.

关键词 计算机取证;Hash;数据完整性;组合编码;有限域

中图法分类号 TP309 **DOI 号**: 10.3724/SP.J.1016.2011.00847

An Efficient Integrity Check Method for Fine-Grained Data over Galois Field

CHEN Long[1] WANG Guo-Yin[2]

[1](*Institute of Computer Forensics, Chongqing University of Posts and Telecommunications, Chongqing* 400065)

[2](*Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing* 400065)

Abstract Fine-grained data integrity checking methods by crossing hashing could isolate a portion of corrupted data segments and assure the integrity of other data at the same time, so as to mitigate the disaster effect on the data by some random errors or intentional forging modification. To improve the efficient of current available method for multi-error cases, a new crossing-hash integrity checking method is proposed based on Galois field uniform partition of multi-dimension structure, herein an efficient integrity indication code for multi-errors case is constructed accordingly. The method has a modular hash check structure. All hashes are divided into several groups, where each group with q rows $d-1$ columns hashes can indicate the integrity of all data independently in a moderate grain and combined hashes of several groups can indicate the integrity of data in a finer grain. At the same time, in a d dimension vector space over $GF(q)$, one more error can be indicated by adding q rows $d-1$ columns hashes every time. Performances with various parameters of the code are analyzed. The performances analysis and experiments results show that this code can indicate multiple errors accurately and efficiently. The code provides a scalable scheme for different applications with several parameters.

Keywords computer forensics; hash; data integrity; combinatorial coding; Galois field

---

收稿日期:2008-10-18;最终修改稿收到日期:2011-01-07.本课题得到国家自然科学基金(60573068)、重庆市自然科学基金项目(CSTC 2008BA2041,2007BB2454)、重庆邮电大学博士启动基金(A2009-25)资助.**陈 龙**,男,1970 年生,博士,教授,主要研究领域为计算机取证、网络安全、智能信息处理.E-mail: chenlong@cqupt.edu.cn.**王国胤**,男,1970 年生,博士,教授,博士生导师,主要研究领域为数据挖掘、粗糙集、粒计算、知识技术、智能信息安全.

1 引 言

由于现实需求的推动,计算机取证研究发展非常迅速,引起了众多研究者的关注[1-2].

Hash 检验的一个重要应用是在进行取证复制的过程中计算并存储取证映像的 Hash 值,从而保证取证分析用的副本的完整性,实现证据固定与保全.取证映像(完全复制件)往往数据量很大,而计算机取证面临的主要难题之一是海量数据处理[3].包括取证映像在内的海量数据的完整性不能只停留在整体是否可靠、未被修改的层面上,因为若有偶然数据变化就会影响全部数据的可用性、可信性.细粒度的数据完整性检验成为支持改善海量数据可用性的一种重要手段,文献[4-5]在开发取证映像工具时基于直观的磁头、柱面、扇区区分法从不同的角度对磁盘数据进行交叉检验,实现了一种基本的细粒度完整性检验方案,增强了磁盘数据的完整性与可用性.一般性的问题可归纳为:如何高效地判断每个细粒度数据对象是否具有完整性? 细粒度是相对于传统关注数据对象大小的概念.例如考虑目录中的单个文件、一个文件的独立分片——小数据块、磁盘中扇区级的物理存储块[6]或数据流中的数据包.这样一来,伴随海量数据处理本身的问题,其完整性检验面临新问题——完整性检验 Hash 数据也成了大规模数据.Hash 检验数据具有随机性,无法使用数据压缩技术进行压缩,对远程取证映像获取时的网络传输效率及完整性检验数据的存储带来较大的影响.例如:一个 512GB 硬盘的扇区级 MD5 Hash 值将需要 16GB 的存储量,如果使用强度较高的 SHA-256 则需要 32GB.

借鉴纠错编码思想[7]可以在低出错率条件下通过交叉完整性检验使用少量 Hash 实现细粒度的完整性检验,从而减少 Hash 数据量[8].相对于每个数据对象使用 1 个 Hash 进行监督的方案,体现为 Hash“压缩”.尽管可以借鉴纠错编码的思想,但研究表明完整性交叉检验涉及的编码在编码设计、编码性质及分析方法上与纠错编码都不相同,因而需要细粒度数据完整性指示编码理论与完整性检验新方法.陈龙等在讨论细粒度数据完整性指示编码基本性质的基础上已分别构造了针对单错、多错的细粒度数据完整性指示编码[8-10].基于复数旋转码构造的复数旋转多错完整性指示码[9]具有指示多错、模块化结构的特点,但它在指示稍多的错误时效率不高.本文考虑多错完整性指示的实际需求,提高编码的效率,基于有限域上的迹函数的性质,提出由有限域上的特殊向量组生成系列扩展迹函数——投影函数,利用系列投影函数可实现有限域的均匀交叉划分,进而构造了高压缩率的多错完整性指示码.

2 完整性指示编码

2.1 完整性检验编码思想

设 X_1,X_2,X_3,X_4,X_5,X_6 表示 6 个数据对象,采用 4 个 Hash 监督这 6 个数据对象,使用如下的监督关系(交叉检验):

$$\begin{cases} X_1 \parallel X_2 \parallel X_4 = h_1 \\ X_1 \parallel X_3 \parallel X_5 = h_2 \\ X_2 \parallel X_3 \parallel X_6 = h_3 \\ X_4 \parallel X_5 \parallel X_6 = h_4 \end{cases} \tag{1}$$

式(1)中的“$\parallel$”表示将数据对象连接成一个数据流,“=”表示将左端的数据流进行单向 Hash 运算,等式右端 h_1,h_2,h_3,h_4 表示 Hash.该监督方案可准确指示一个错误.在需要进行完整性检验时采用相同顺序处理数据对象,按式(1)重新生成 Hash,与事先存储的 Hash 进行比较以判断数据对象是否发生改变.例如 h_1,h_2 与其原值不相符时,无法证明 X_1 具有完整性,判定 X_1 出错,而 h_3,h_4 无变化可明确表明其它数据对象没有出错,具有完整性.

为方便分析编码的性质,将式(1)的监督关系表达为一个监督矩阵 $\boldsymbol{A}[m,n]$,如表 1 所示.

表 1 监督矩阵实例

		$\boldsymbol{A}$					
		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
m	1	1	1	0	1	0	0
	2	1	0	1	0	1	0
	3	0	1	1	0	0	1
	4	0	0	0	1	1	1

细粒度完整性检验需要从所有数据对象中指示出不具有完整性的若干具体数据对象,把基于纠错编码思想的完整性检验方法设计称为完整性指示编码,其基本原则是不能将出错数据对象判定为正常数据对象.

2.2 完整性指示码的基本概念

定义 1(完整性指示码). 设需要检验完整性的数据对象有 n 个,若存在一种监督关系,使得生成并存储的 m 个 Hash 值,在对 n 个对象进行检验时能准确指示任意的 t 个出错对象,但无法准确指示

当 $n\geq t+1$ 时某组合中存在的 $t+1$ 个错误，其中受监督次数最多的某个数据对象受到 k 个 Hash 监督 ($k\geq 1$). 这种监督方案称为完整性指示码，记为 $C=[n,m,t,k]$.

定义 2.　完整性指示码的压缩率 η 为数据对象数 n 和使用 Hash 值个数 m 之比.

定义 3.　利用完整性指示码 $C=[n,m,t,k]$ 进行完整性检验时，若实际出现的错误数量 e 大于编码设计时可准确指示的错误数量 t，则可能出现将正常数据对象判定为出错数据对象的情况，即指示出错的对象数量大于 e. 这种现象称错误放大. 错误数据对象的组合关系不同会导致实际指示错误数量也可能不同. 考察 e 个数据错误对象的所有组合，可得其平均值. 实际指示错误数量的平均值和实际出错数量的比值称错误放大率，记为 $\beta(e)$. 由于码 C 能准确指示 t 个错误，因此，出现 $t+1$ 个错误时最能体现 C 的基本特性，将 $e=t+1$ 时的 $\beta(e)$ 简记为 β，称为 C 的基准错误放大率.

特别地，规定 $\beta(0)=1$.

3　有限域多错完整性指示码

复数旋转完整性指示码实现了有限域二维结构上不同方向之间的均匀交叉，可指示多个错误，不同的监督方式有 $q+1$ 种[9]. 分析发现，不同的监督方式和有限域 $GF(q)$ 上 $q+1$ 个二维向量构成的两两线性无关的向量组对应. 鉴于复数旋转完整性指示码的压缩率不够高，我们进一步讨论有限域 $GF(q)$ 上三维及多维结构形成的均匀交叉监督方案. 下文涉及有限域元素的运算均限定在有限域上，其它运算则为普通加法、乘法(如指数的加、乘).

3.1　基本概念

定义 4.　设 q 是素数或素数幂(下同)，则有限域 $GF(q^d)$ 在 $GF(q)$ 上的迹函数[11] $Tr:GF(q^d)\rightarrow GF(q)$ 为

$$Tr(u)=u+u^q+u^{q^2}+\cdots+u^{q^{d-1}} \tag{2}$$

把 $Tr(u)$ 称为 u 在 $GF(q)$ 上的迹. 该迹函数是一个线性函数，对任意的 $u,v\in GF(q^d)$，有以下 5 条性质[11]：

① $Tr(u+v)=Tr(u)+Tr(v)$；

② 对任意的 $a\in GF(q)$，$Tr(au)=aTr(u)$；

③ $Tr(u^q)=Tr(u)$；

④ 映射 $Tr:GF(q^d)\rightarrow GF(q)$ 是满射；

⑤ 对每一 $a\in GF(q)$，恰有 q^{d-1} 个元素的迹为 a.

设 α 是有限域 $GF(q^d)$ 关于 $GF(q)$ 的本原元，任一元素 $u\in GF(q^d)$，则 u 可由自然基 $1,\alpha,\cdots,\alpha^{d-1}$ 表示为 $u=c_{d-1}\alpha^{d-1}+\cdots+c_1\alpha+c_0$，其中 $c_i\in GF(q)$. 由①、②线性性质有

$$Tr(u)=c_{d-1}Tr(\alpha^{d-1})+\cdots+c_1Tr(\alpha)+c_0Tr(1) \tag{3}$$

因此，先计算出一个自然基中所有元素的迹，由这些元素在 $GF(q)$ 上的线性组合即可得所有元素的迹.

定义 5.　向量组 $\boldsymbol{V}$ 中有 γ 个 $GF(q)$ 上的 d 维非零向量 ($d>1$)，如果 $\boldsymbol{V}$ 中的任意 d 个向量都线性无关，则称 $\boldsymbol{V}$ 是 $GF(q)$ 上的 d-线性无关向量组，向量个数 γ 称为向量组 $\boldsymbol{V}$ 的阶.

Rizzo 提出了采用系统码矩阵(systematic code matrix)结合范德蒙德矩阵(Vandermonde matrix)的方式来构成线性编码向量组[12]，其编码向量数可以达到 $(d-1)+q$，但是却无法保证其中的任意 d 个向量均线性无关. 陶均等在设计数据分散编码存储方案时讨论了线性分组编码需要的编码向量的构造问题，根据其结论可知 $GF(2)$ 及其扩张域上 d-线性无关向量组 $\boldsymbol{V}$ 的阶 γ 的最大值的理论上界和下界[13]. 采用和陶均等人相同的思路和方法，可知阶 γ 的最大值的理论上、下界对于任意的 $GF(q)$ 上的 d-线性无关向量组 $\boldsymbol{V}$ 也成立. 若同时考虑另一因素——维度 d，可以得到 $d>q$ 时的更紧的下界，所以有结论如下.

定理 1.　设 q 是素数或素数幂，则 $GF(q)$ 上的 d-线性无关向量组 $\boldsymbol{V}$ 的阶 γ 的最大值满足如下的上、下界：

$$\max(q+1,d+1)\leq\gamma\leq q+d-1 \tag{4}$$

其中 $d>1$，max 表示两者取大.

证明.　设 α 是 $GF(q)$ 的本原元，则 $GF(q)$ 上的所有元素均可以表示为 $\{0,1,\alpha,\alpha^2,\cdots,\alpha^{q-2}\}$.

首先，采用数学归纳法证明关于 γ 上界的不等式.

(1) 当 $d=2$ 时，选取向量集合 $\boldsymbol{V}=\{(1,0),(1,1),(1,\alpha),\cdots,(1,\alpha^{q-2}),(0,1)\}$，共计 $q+1$ 个元素. 考察有限域上的其它 2 维向量，可将它们划分为 $\{\alpha^x,\alpha^y\}$、$\{\alpha^x,0\}$、$\{0,\alpha^y\}$ 三类，其中 $x,y\neq 0$. 那么，

$$\begin{cases}(\alpha^x,\alpha^y)=(\alpha^x\cdot\alpha^0,\alpha^x\cdot\alpha^{y-x})=\alpha^x\cdot(1,\alpha^{y-x})\\(\alpha^x,0)=(\alpha^x\cdot\alpha^0,\alpha^x\cdot 0)=\alpha^x\cdot(1,0)\\(0,\alpha^y)=(\alpha^y\cdot 0,\alpha^y\cdot\alpha^0)=\alpha^y\cdot(0,1)\end{cases} \tag{5}$$

所以，这三类向量中的任何一个向量都可由向量组 $\boldsymbol{V}$ 中的某一个线性表示. 所以 $d=2$ 时 d-线性无关向量组 $\boldsymbol{V}$ 刚好有 $q+1$ 个向量两两线性无关，不

等式成立.

(2)假设不等式 $\gamma\leq q+d-1$ 对于 $d=k(k>1)$ 时均成立,证明 $d=k+1$ 时的情况.假设不等式在 $d=k+1$ 时不成立,则有假设命题:可以得到一个 $k+1$ 维向量组 $\boldsymbol{V}'$,其向量数至少为 $q+k+1$ 且其中任意 $k+1$ 个向量都线性无关.

任取 $\boldsymbol{V}'$ 中一个 $k+1$ 维向量,与 $\boldsymbol{V}'$ 中的其余向量作高斯消元,使其余 $q+k$ 个向量消去一维,则高斯消元后的 $q+k$ 个 k 维向量,应满足任意 k 个向量都线性无关,而这与结论 $d=k$ 时 $\gamma\leq q+k-1$ 相矛盾.

因此,假设命题不成立,不等式在 $d=k+1$ 时仍成立.

(3)综上所述,由数学归纳法原理可知,对一切 $d>1$,关于 γ 上界的不等式均成立.

然后,采用构造法证明关于 γ 下界的不等式.

若 $q\geq d$,参考范德蒙德(Vandermonde)矩阵,增加两个特殊向量,由 $GF(q)$ 中的元素 $0,1,\alpha,\alpha^2,\cdots,\alpha^{q-2}$ 构造包含 $q+1$ 个向量的向量组 $\boldsymbol{V}$ 及展开的矩阵如下:

$$\boldsymbol{V}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \boldsymbol{v}_3\\ \vdots\\ \boldsymbol{v}_{q-1}\\ \boldsymbol{v}_q\\ \boldsymbol{v}_{q+1}\end{bmatrix}=\begin{bmatrix}(\alpha^0)^0 & (\alpha^0)^1 & (\alpha^0)^2 & \cdots & (\alpha^0)^{d-1}\\ (\alpha^1)^0 & (\alpha^1)^1 & (\alpha^1)^2 & \cdots & (\alpha^1)^{d-1}\\ (\alpha^2)^0 & (\alpha^2)^1 & (\alpha^2)^2 & \cdots & (\alpha^2)^{d-1}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ (\alpha^{q-2})^0 & (\alpha^{q-2})^1 & (\alpha^{q-2})^2 & \cdots & (\alpha^{q-2})^{d-1}\\ 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{6}$$

分析向量组 $\boldsymbol{V}$ 中的任意 d 个向量的线性无关性,分 3 种情况讨论:

(1)从 $\boldsymbol{V}$ 中任选 d 个向量但不包括 $\boldsymbol{v}_q$、$\boldsymbol{v}_{q+1}$ 时,此时所得矩阵即为 $d\times d$ 的范德蒙德矩阵.显然对应的范德蒙德行列式不等于 0,此时,任选的 d 个向量线性无关.

(2)从 $\boldsymbol{V}$ 中任选 d 个向量,其中包括 $\boldsymbol{v}_q$ 或 $\boldsymbol{v}_{q+1}$ 之一时,若 $d=2$,显然所得矩阵对应行列式不等于 0,否则用 $\boldsymbol{v}_q$ 或 $\boldsymbol{v}_{q+1}$ 对其它 $d-1$ 个向量进行高斯消元,必要时每行提取公因子,可得 $(d-1)\times(d-1)$ 的范德蒙德矩阵.同理,所选的 d 个向量线性无关.

(3)从 $\boldsymbol{V}$ 中任选 d 个向量,其中包括 $\boldsymbol{v}_q$ 和 $\boldsymbol{v}_{q+1}$ 时:若 $d=2$,则只有 $\boldsymbol{v}_q$ 和 $\boldsymbol{v}_{q+1}$,结论成立;若 $d=3$,另一个向量的分量均不为 0,显然此时 3 个向量线性无关;否则用 $\boldsymbol{v}_q$ 和 $\boldsymbol{v}_{q+1}$ 对其它 $d-2$ 个向量进行高斯消元,然后每行提取公因子,可得 $(d-2)\times(d-2)$ 的范德蒙德矩阵.同理,任选的 d 个向量线性无关.

综合这 3 种情况可知所构造的向量组 $\boldsymbol{V}$ 中的任意 d 个向量都线性无关.即 $q\geq d$,对任意的 $GF(q)$ 都存在 $\gamma=q+1$ 的 d-线性无关向量组,即 $q+1\leq\gamma$ 成立.

若 $q<d$,构造包含 $d+1$ 个向量的向量组 $\boldsymbol{V}$ 及展开的矩阵如下:

$$\boldsymbol{V}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \vdots\\ \boldsymbol{v}_d\\ \boldsymbol{v}_{d+1}\end{bmatrix}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & 1\\ 1 & 1 & \cdots & 1\end{bmatrix} \tag{7}$$

显然,向量组 $\boldsymbol{V}$ 中的任意 d 个向量都线性无关,即 $d+1\leq\gamma$ 也成立.

所以,式(4)中左端的不等式成立.综合起来,定理成立. 证毕.

定理 2. 有限域 $GF(q)$ 上 d-线性无关向量组 $\boldsymbol{V}$ 的阶 γ 在以下特例可达到上界.特例 1:$q=2$;特例 2:$d=2$;特例 3:$d=3$ 且 $GF(q)$ 的特征为 2.

证明. 特例 1 和特例 2 中的上下界相等,所以 γ 可达到上界;特例 3 采用构造法证明.特例 3 中,q 为 2 的幂次,向量组 $\boldsymbol{V}$ 构造如下:

$$\boldsymbol{V}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\\ \boldsymbol{v}_3\\ \vdots\\ \boldsymbol{v}_{q-1}\\ \boldsymbol{v}_q\\ \boldsymbol{v}_{q+1}\\ \boldsymbol{v}_{q+2}\end{bmatrix}=\begin{bmatrix}1 & 1 & 1\\ 1 & \alpha^1 & (\alpha^1)^2\\ 1 & \alpha^2 & (\alpha^2)^2\\ \vdots & \vdots & \vdots\\ 1 & \alpha^{q-2} & (\alpha^{q-2})^2\\ 1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix} \tag{8}$$

由定理 1 的证明过程中构造的向量组可知,从 $\boldsymbol{V}$ 中任选 3 个向量不包括 $\boldsymbol{v}_{q+2}$ 时 3 个向量线性无关,只需额外证明选中了 $\boldsymbol{v}_{q+2}$ 的情况.设已选中 $\boldsymbol{v}_{q+2}$ 为 3 个向量之一,选中的另外两个向量为 $\boldsymbol{v}_i,\boldsymbol{v}_j$,其中 $i<j$.分 3 种情况讨论:如果 $q\leq i<j$,则可构成单位矩阵,结论显然;如果 $i<q\leq j$,由于 $\boldsymbol{v}_i$ 的分量均不等于零,显然,所选向量的行列式不等于零;如果 $i<j<q$,则所选向量的行列式等于 $\alpha^{2i-2}-\alpha^{2j-2}$,由于 $i<j$,所以 $2i-2<2j-2$,又由于 q 是偶数,所以 $2i-2\neq 2j-2-q-1$,于是 $\alpha^{2i-2}\neq\alpha^{2j-2}$.所以,所选向量的行列式不等于零,所选的 3 个向量线性无关.所以,此时 γ 可达定理 1 中的上界 $q+2$. 证毕.

3.2 有限域划分

定义 6(投影函数). 设 q 是素数或素数幂，从 $GF(q)$ 上 d-线性无关向量组 $\boldsymbol{V}$ 中选取一个向量 $\boldsymbol{v}=(a_0,a_1,\cdots,a_{d-1})$，$\forall a_i\in GF(q)$，令 $GF(q^d)$ 的自然基元素 $1,\alpha,\cdots,\alpha^{d-1}$ 在 $GF(q)$ 上的投影构成向量 $\boldsymbol{v}$，即 $\pi_v(\alpha^i)=a_i$，$i=0,1,\cdots,d-1$，定义 $GF(q^d)$ 中任一元素 u 在 $GF(q)$ 上关于向量 $\boldsymbol{v}$ 的投影为

$$\begin{aligned}\pi_v(u)&=c_{d-1}\pi_v(\alpha^{d-1})+\cdots+c_1\pi_v(\alpha)+c_0\pi_v(1)\\&=c_{d-1}a_{d-1}+\cdots+c_1a_1+c_0a_0\end{aligned}\tag{9}$$

投影函数满足迹函数的性质中除③外其它 4 条，称向量 $\boldsymbol{v}$ 为投影参考向量.

由 $GF(q^d)$ 中投影相同的元素构成一个集合，于是每个投影函数都可以得到 $GF(q^d)$ 在 $GF(q)$ 上的一个均匀划分——分为 q 个包含 q^{d-1} 个元素的集合，由定理 1，$GF(q^d)$ 关于 $GF(q)$ 的不同划分至少可达 $q+1$ 种或 $d+1$ 种.

不同投影函数得到的划分之间形成均匀交叉，并有如下定理.

定理 3. $GF(q^d)$ 上的一个元素 x 与 x 在 $GF(q)$ 上任意 d 个投影函数的一组投影值 $b_1,b_2,\cdots,b_d$ 可相互唯一确定.

证明. 设所选用的投影参考向量分别为 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_d$，$d$ 个投影函数为 $\pi_{v_1},\pi_{v_2},\cdots,\pi_{v_d}$，由投影函数可知 x 有确定的 d 个投影值. 反之，若已知投影值 $b_1,b_2,\cdots,b_d$，则 $x=(x_1,x_2,\cdots,x_d)$ 满足如下方程(所有运算都在有限域上进行)：

$$\begin{cases}\pi_{v_1}(x)=a_{11}x_1+a_{12}x_2+\cdots+a_{1d}x_d=b_1\\\pi_{v_2}(x)=a_{21}x_1+a_{22}x_2+\cdots+a_{2d}x_d=b_2\\\qquad\vdots\\\pi_{v_d}(x)=a_{d1}x_1+a_{d2}x_2+\cdots+a_{dd}x_d=b_d\end{cases}\tag{10}$$

用 $\boldsymbol{A}=(a_{ij})$ 表示式(10)中的系数矩阵，因投影参考向量 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_d$ 线性无关，所以 $\det\boldsymbol{A}\neq0$，则 x 有唯一解. 利用行列式表示形式，x 的分量分别为

$$x_1=\frac{\det\boldsymbol{A}_1}{\det\boldsymbol{A}},x_2=\frac{\det\boldsymbol{A}_2}{\det\boldsymbol{A}},\cdots,x_d=\frac{\det\boldsymbol{A}_d}{\det\boldsymbol{A}}\tag{11}$$

其中，$\det\boldsymbol{A},\det\boldsymbol{A}_1,\det\boldsymbol{A}_2,\cdots,\det\boldsymbol{A}_d$ 分别为式(10)左端系数行列式和将第 1 列，第 2 列，…，第 d 列分别替换成式(10)右端投影值列所得的行列式. 证毕.

定理 3 表明可由特定元素在不同划分中的投影值对其进行定位.

3.3 有限域多错完整性指示码

定义 7(有限域多错完整性指示码). 设完整性检验数据对象有 $n=q^d$ 个，其中 q 是素数或素数幂，d 是大于 1 的整数，需要准确指示数据对象中的任意 t 个错误对象，将 q^d 个数据对象分别对应于 $GF(q)$ 上的一个 d 维向量. 从 $GF(q)$ 上的 d-线性无关向量组 $\boldsymbol{V}$ 中任意选取 $k=(d-1)t+1$ 个投影参考向量，将关于函数 π_{v_j} $(j=1,2,\cdots,k)$ 投影相同的所有数据对象用一个 Hash 监督. 于是，每个函数都把 n 个数据对象均匀划分为 q 份分别进行监督，参与每个 Hash 计算的数据对象有 q^{d-1} 个，此监督方案共有 kq 个 Hash 值，构造得到有限域多错完整性指示码 $C=[n,m,t,k]=[q^d,kq,t,k]$，$k=(d-1)t+1$. 简称有限域多错指示码，简记为 GFIIC (Galois Field multi-error Integrity Indication Code).

有限域多错完整性指示码在 $d=2$ 时为 $C=[q^2,(t+1)q,t,t+1]$，此时的特例为复数旋转完整性指示码[9]. 当 $d=3$ 时，有限域多错完整性指示码为 $C=[q^2,(t+1)q,t,t+1]$，称为有限域立方指示码，简记为 GFC.

定理 4. 有限域多错完整性指示码 $C=[n,m,t,k]=[q^d,kq,t,k]$，$k=(d-1)t+1$ 的 k 组共 kq 个 Hash 可准确指示任意的 t 个错误.

证明. 已知每个数据受到 k 个 Hash 监督. 设任意的 t 个出错数据对象对应于 t 个 $GF(q)$ 上的向量 $\boldsymbol{w}_i=(w_{i1},w_{i2},\cdots,w_{id})$，$i=1,2,\cdots,t$. 另设一未出错数据对象对应于向量 $\boldsymbol{w}_x=(w_{x1},w_{x2},\cdots,w_{xd})$. 那么任一 $\boldsymbol{w}_i$ 至多与 $\boldsymbol{w}_x$ 同时受 $d-1$ 个 Hash 监督(若 $\boldsymbol{w}_i$ 与 $\boldsymbol{w}_x$ 同受某个 Hash 监督，则根据该 Hash 无法判断 $\boldsymbol{w}_x$ 是否出错).

否则，假设 $\boldsymbol{w}_i$ 与 $\boldsymbol{w}_x$ 同受 d 个 Hash 监督，则该码使用的 k 个投影函数中存在 d 个投影函数，使得 $\boldsymbol{w}_i$ 与 $\boldsymbol{w}_x$ 的投影相同，即有

$$\begin{cases}\pi_{v_1}(\boldsymbol{w}_x)=b_1=\pi_{v_1}(\boldsymbol{w}_i)\\\pi_{v_2}(\boldsymbol{w}_x)=b_2=\pi_{v_2}(\boldsymbol{w}_i)\\\qquad\vdots\\\pi_{v_d}(\boldsymbol{w}_x)=b_d=\pi_{v_d}(\boldsymbol{w}_i)\end{cases}\tag{12}$$

由定理 3 得 $\boldsymbol{w}_x=\boldsymbol{w}_i$. 与已知的两者不相同矛盾. 即假设不成立. 所以，任一 $\boldsymbol{w}_i$ 至多与 $\boldsymbol{w}_x$ 同受 $d-1$ 个 Hash 监督.

根据鸽巢原理，监督 t 个出错对象的 Hash 且监督 $\boldsymbol{w}_x$ 的 Hash 至多有 $(d-1)t$ 个 Hash 监督，即最多在 $(d-1)t$ 个 Hash 上 $\boldsymbol{w}_x$ 无法与出错对象 $\boldsymbol{w}_i$ 区分，$i=1,2,\cdots,t$. $(d-1)t+1$ 个 Hash 中的另一个可指示 $\boldsymbol{w}_x$ 具有完整性(未出错).

所以，kq 个 Hash 可准确指示任意的 t 个错误数据对象，而其它的数据对象具有完整性. 证毕.

由码 GFIIC 的完整性检验、监督关系可知，每个分组均可独立指示全部数据的完整性. 同时，由码中 k 与 t 的关系及定理 4 的证明过程可见，每增加 $(d-1)$ 组共 $(d-1)q$ 个 Hash 即可多指示一个错. 所以，码 GFIIC 体现出模块化的 Hash 结构.

3.4 GFIIC 码 Hash 生成与检验

3.4.1 Hash 生成算法

数据对象监督关系. 设数据对象个数为 n，指示错误能力为 t. 选取适当的素数或素数幂 q，及维数 d，使得 $n \leq q^d$ 且 $k=(d-1)t+1 \leq \gamma$. 从有限域 $GF(q)$ 上的 d-线性无关向量组中选取 k 个向量. 把 k 个投影函数所生成的 Hash 看成 q 行 k 列的 Hash 矩阵. 将数据对象对应到有限域的元素：把数据对象从 0 到 $n-1$ 依次编号，将每个编号 $j(j=0,1,\cdots,n-1)$ 表示为 $GF(q)$ 上的 d 维向量 $(r_d,\cdots,r_2,r_1)$，满足 $j=(r_d,\cdots,r_2,r_1)=r_d q^{d-1}+\cdots+r_2 q+r_1$. 对数据对象进行投影划分：计算数据对象 $j(j=0,1,\cdots,n-1)$ 在第 l 个 $(l=1,2,\cdots,k)$ 向量（记 $\boldsymbol{v}_l=(a_d,\cdots,a_2,a_1)$）上的投影 u，即 $u=\pi_{\boldsymbol{v}_l}(j)=a_d r_d+\cdots+a_2 r_2+a_1 r_1$，依据投影 u 判定数据对象 j 参与第 $u+1$ 行第 l 列的 Hash 计算，即同一投影函数下投影相同的元素受同一个 Hash 监督. 每个 Hash 由同一投影函数下投影相同的 q^{d-1} 个数据对象计算得到. 每个投影函数可生成 q 个 Hash.

并发计算模式. 上述 Hash 值生成时依次读入数据对象，可采用并发计算模式[8]. 即并发计算所有该对象参与计算的 k 个 Hash 值，将中间结果暂存起来，后面某个数据对象读入之后取出该中间结果继续进行. 并发计算方式的源数据只需读入一次，可有效缓解大数据量的 I/O 瓶颈问题，每个数据对象参与了 Hash 计算 k 次. Hash 计算数据总量为 knB. B 为数据对象的平均大小，假设其为 512 字节的整数倍.

再 Hash 计算模式. 可信计算环境存储器数据完整性检验使用了 Hash 树或 Merkle 树，其中应用了再 Hash 机制[14-15]. 应用再 Hash 机制可加快细粒度完整性检验中的 Hash 计算速度. 先分别计算每个数据对象的 Hash，通过一定程度的缓冲，把受同一个 Hash 监督的元素对应的 Hash 数据连接后再进行 Hash 运算，即用完整性数据 $H(H(D_1)\|H(D_2)\|\cdots\|H(D_q))$ 替代 $H(D_1\|D_2\|\cdots\|D_q)$. 其中 H 表示完整性检验单向 Hash 函数，$\|$ 表示数据连接运算，D_i 为数据对象. 再 Hash 方法计算数据总量为 $\left(1+\frac{k}{32}\cdot\frac{512}{B}\right)nB$. 其算法如下.

再 Hash 计算算法.

输入：码 $[n,m,t,k]$，n 个数据对象，d-线性无关向量组

输出：Hash 矩阵

算法步骤：

1. 根据有限域 $GF(q)$、维数 d 以及指示错误能力 t 值从准备好的相应 d-线性无关向量组中取出 $k=(d-1)\times t+1$ 个向量；

2. 初始化 Hash 矩阵，$j=0$；

3. 建立和 Hash 矩阵对应的缓冲数据矩阵；

4. 读入第 j 个数据对象；

5. 计算该数据对象的 Hash 数据 h_j；

6. 由 k 个对应的投影函数计算该数据对象的监督关系，确定监督该数据对象的 k 个 Hash 以及相应的 Hash 数据缓冲位置；

7. 将 Hash 数据 h_j 分别存入缓冲数据矩阵中的对应 k 个位置（已有缓冲数据之后）；

8. 如果 k 个位置中某个或某几个位置的数据达到合适的规模（如 32 字节、512 字节等），分别取出 Hash 矩阵中对应中间结果或初值计算缓冲数据的对应 Hash，将中间结果再存入 Hash 矩阵，并把缓冲数据清除；否则继续；

9. $j=j+1$，重复步 4～8 直到数据对象处理完；

10. 输出 Hash 矩阵.

3.4.2 Hash 检验算法

Hash 检验算法.

输入：Hash 矩阵，码 $[n,m,t,k]$，n 个数据对象，d-线性无关向量组

输出：出错数据对象编号

算法步骤：

1. 选取原 Hash 矩阵生成时相同的参数，采用相同的 Hash 计算方式生成 Hash 矩阵；

2. 将所得 Hash 矩阵和原 Hash 矩阵进行比较，如果任意一组（列）Hash 完全相符，则没有出错，所有数据都具有完整性，结束. 否则任意一组中至少有一个 Hash 不相符，继续下一步；

3. 先检查 d 个向量生成的 Hash 组，统计不相符的 Hash 数量；

4. 设这 d 组 Hash 分别有 $x_d,\cdots,x_2,x_1$ 个 Hash 不相符，则最多有 $x_d*\cdots*x_2*x_1$ 个数据对象出错；

5. 从 Hash 矩阵相应列 x_d 个 Hash 中取 1 个得到其行号（从小到大）r_d，同理取得 $r_{d-1},\cdots,r_2,r_1$. 由定理 3，用步 3 中的 d 个向量和向量 $(r_d-1,\cdots,r_2-1,r_1-1)$ 可求得对应的数据对象，其编号记为 j；依据数据对象监督关系依次检查数据对象 j 在其它的 $(d-1)t-d+1$ 组 Hash 中对应的 Hash 是否相符，若全都不相符，则数据对象 j 出错——不具有完整性，输出编号 j；

6. 重复步 5 验证其它数据对象是否出错.

3.5 性能分析

(1) 参数边界分析. 采用有限域多错指示码，以定理 1 中的 γ 下界为 d-线性无关组的向量数. 若固定指示错误数 t 和数据对象 $n=q^d$，此时压缩率 η 上界为

$$\eta=\frac{n^{1-1/d}}{(d-1)t+1},\ (d-1)t\leq n^{1/d} \tag{13}$$

设维数 d 小于有限域的阶 q(否则问题简化，只能指示单错)，则 $GF(q)$ 上的有限域多错指示码指示错误能力 t 的上界为

$$t\leq\left\lfloor\frac{q}{d-1}\right\rfloor \tag{14}$$

若固定指示错误能力 t 时，压缩方案维数的上界：

$$d\leq\left\lfloor\frac{q}{t}\right\rfloor+1 \tag{15}$$

(2) 压缩率分析. 有限域多错指示码的压缩率为

$$\eta=\frac{1}{(d-1)t+1}q^{d-1} \tag{16}$$

显然，对某一固定维数和固定数量的数据对象，需要准确指示的错误越多，可实现的压缩率就越低. 对某一固定错误数 t 和固定维数 d，q 越大时检验数据对象越多，压缩率也就越高.

对某一固定错误数 t 和相同的 Hash 个数 m，假设采用 $d+1$ 维方案的 Hash 数据量 $m=(dt+1)q$ 与 d 维方案使用的 Hash 数据量相同，则 $d+1$ 维和 d 维方案的压缩率之比为

$$\Delta_d=\frac{\eta}{\eta_1}=\frac{\dfrac{q^d}{dt+1}}{\dfrac{q_1^{d-1}}{(d-1)t+1}}=\frac{((d-1)t+1)^d}{(dt+1)^d}q \tag{17}$$

而对于某一固定错误数 t 和检验对象总数 $n=q^{d+1}$，采用 $d+1$ 维和 d 维方案的压缩率之比为

$$\Delta'_d=\frac{\eta}{\eta'_1}=\frac{\dfrac{q^d}{dt+1}}{\dfrac{q_1^{d-1}}{(d-1)t+1}}=\frac{(d-1)t+1}{dt+1}q^{\frac{1}{d}} \tag{18}$$

由式(15)表达的维数限制可见，一般而言维数高的方案压缩率更高(不是所有参数都能同时满足不同维度参数关系).

(3) 错误放大率. 依据指示码的指示错误能力，有限域多错指示码的维数等参数分别讨论如下：

① 当 $t=1$ 时，该码退化为超方体单错指示码，其超方体的阶 r 只能取素数或素数幂，错误放大率增长较快，其具体性能参见文献[10].

② 当 $d=2,t\geq2$ 时，错误放大率增长稍快，其具体性能参见文献[9]讨论的复数旋转完整性指示码.

③ 当 $d=3,t\geq2$ 时，用抽样的方式计算出不同有限域下有限立方指示码的基准错误放大率 β. 实验设定的抽样次数为 20 万次. 得到 GFC 指示码的基准错误放大率如表 2 所示.

表 2　GFC 码的基准错误放大率

有限域	错误放大率						
	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
$GF(4)$	1.62	—	—	—	—	—	—
$GF(5)$	1.50	—	—	—	—	—	—
$GF(7)$	1.33	1.13	—	—	—	—	—
$GF(8)$	1.28	1.07	1.03	—	—	—	—
$GF(9)$	1.23	1.07	1.03	—	—	—	—
$GF(11)$	1.17	1.02	1.01	1.00	—	—	—
$GF(13)$	1.13	1.02	1.00	1.00	1.00	—	—
$GF(16)$	1.09	1.01	1.00	1.00	1.00	1.00	1.00
$GF(17)$	1.08	1.01	1.00	1.00	1.00	1.00	1.00
$GF(19)$	1.07	1.00	1.00	1.00	1.00	1.00	1.00
$GF(23)$	1.05	1.00	1.00	1.00	1.00	1.00	1.00

④ 当 $d=3,t\geq2$ 时，以 $q=16$ 为实例，考察有限域立方指示码的一般错误放大率. 通过抽样方法测试产生一定错误数量时的实际指示错误数，由于错误增多，组合关系更多，分别抽样 50 万次，$GF(16)$ 上的 GFC 码的一般错误放大率特性如图 1 所示.

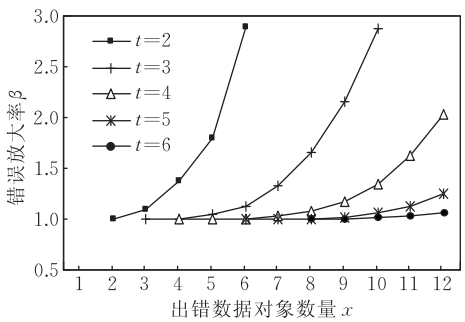

图 1　$GF(16)$ 上的 GFC 码的一般错误放大率

由表 2 和图 1 可见，GFC 指示码 q 很小及 $t=2$ 时有稍明显的出错放大，其它情形的基准错误放大率都很低，随着错误数的增长，t 较小时错误放大率增长稍快，t 较大时错误放大率增长较慢. 对比定理 4 的结论可知，GFC 在设计指示错误能力范围内能完全准确指示出任意的多个错误，对于少量超出此范围的错误，大部分情况下也可以正确指示，其它情况类似.

(4) 时间性能. 以有限域立方指示码为例，测试码的时间性能. 随机生成 2M、4M、8M 大小的文件各 10 个，使用 MD5 Hash 函数测试有限域立方码采用并发计算方式和再 Hash 方式时相对于单 Hash 生成的时间性能. 设定数据对象分块大小为

512 字节，每组为 4096 块，即 $n=4096$，3 种规格的文件分别有 1 组、2 组、3 组数据对象. 分别采用两种方式生成 Hash 矩阵，另单独生成单一 Hash，测试实际运行时间. 计算 Hash 生成的平均时间，换算为相对于单 Hash 生成所费时间的比值. 有限域立方指示码 Hash 生成的相对时间见表 3.

表 3 有限域立方指示码 Hash 生成的相对时间

计算模式	文件大小	相对时间						
		$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
并发计算模式	2M	1.109	1.149	1.194	1.233	1.275	1.315	1.353
	4M	1.112	1.152	1.191	1.232	1.273	1.312	1.353
	8M	1.093	1.134	1.177	1.218	1.260	1.299	1.341
再Hash模式	2M	1.035	1.047	1.054	1.068	1.072	1.078	1.086
	4M	1.031	1.040	1.050	1.057	1.064	1.074	1.081
	8M	1.039	1.047	1.056	1.064	1.074	1.082	1.090

由表 3 可见，交叉检验带来了 Hash 计算的部分重复，相对于单 Hash 生成时间，生成 Hash 数据所需的时间随着指示错误数的增多呈线性增长. 同时，并发计算方式时间增长稍快，而采用再 Hash 方式时间增长很慢. 再 Hash 方式与传统的生成单一 Hash 方式时间性能差异较小.

4 结 论

细粒度数据完整性检验方法在实现完整性检验的同时可以对少数错误进行准确和高效的隔离，本文针对需要隔离多个错误时现有方案效率不高的问题，提出了多维结构下基于有限域均匀划分的完整性交叉检验方法，具体内容包括：提出了有限域上系列扩展迹函数——投影函数的构造方法，基于系列投影函数实现对有限域的多种均匀交叉划分，构造了有限域上高效的多错完整性指示编码，设计了再 Hash 计算模式以加快 Hash 生成. 该码在设计指示错误能力范围内能准确指示出多个错误，对于少量超出此范围的错误，大部分情况下也可以正确指示. 该码具有模块化的 Hash 结构：由任一组 Hash 在某一中间粒度均可独立指示所有数据的完整性，多组 Hash 结合起来则在更小的基本粒度指示数据的完整性；同时，对于 $GF(q)$ 上的 d 维向量空间，每增加 $(d-1)$ 组共 $(d-1)q$ 个 Hash 即可多指示一个任意错. 分析和实验结果表明该码效率高，可实现几十倍的“Hash 压缩”，具有灵活的参数选择，可满足各种应用的不同需要，对实际应用具有指导作用.

致 谢 感谢论文审稿专家给予的启发性意见、建议，这对提高本文质量起到了重要作用！

参 考 文 献

[1] Wang Ling, Qian Hua-Lin. Computer forensics and its future trend. Journal of Software, 2003, 14(9): 1635-1644(in Chinese)
(王玲，钱华林. 计算机取证技术及其发展趋势. 软件学报，2003，14(9)：1635-1644)

[2] Ding Li-Ping, Wang Yong-Ji. Study on relevant law and technology issues about computer forensics. Journal of Software, 2005, 16(2): 260-275(in Chinese)
(丁丽萍，王永吉. 计算机取证的相关法律技术问题研究. 软件学报，2005，16(2)：260-275)

[3] Golden G. Richard III and vassil roussev. Next-generation digital forensics. Communications of the ACM, 2006, 49(2): 76-80

[4] Jiang Zoe L, Hui Lucas C K, Chow K P, Yiu S M, Lai Pierre K Y. Improving disk sector integrity using 3-dimension hashing scheme//Proceedings of the 2007 International Workshop on Forensics for Future Generation Communication Environments(F2GC-07). 2007: 141-145

[5] Jiang Zoe L, Hui Lucas C K, Yiu S M. Improving disk sector integrity using K-dimension hashing//Proceedings of the International Federation for Information Processing, Advances in Digital Forensics IV. Indrajit Ray, Sujeet Shenoi, 2008, 285: 87-98

[6] Roussev Vassil, Chen Yixin, Bourg Timothy, Richard III Golden G. md5bloom: Forensic filesystem hashing revisited. Digital Investigation, 2006, 3(s1): 82-90

[7] Hamming R W. Error detecting and error correcting codes. The Bell System Technical Journal, 1950, XXVI(2): 147-160

[8] Chen Long, Wang Guo-Yin. An integrity check method for fine-grained data. Journal of Software, 2009, 20(4): 902-909(in Chinese)
(陈龙，王国胤. 一种细粒度数据完整性检验方法. 软件学报，2009，20(4)：902-909)

[9] Chen Long, Fang Xin-Lei, Wang Guo-Yin. Integrity check method for fine-grained data based on complex rotary codes. Journal of Southwest Jiaotong University, 2009, 44(5): 667-671(in Chinese)
(陈龙，方新蕾，王国胤. 基于复数旋转码的细粒度数据完整性指示方法. 西南交通大学学报，2009，44(5)：667-671)

[10] Chen Long, Fang Xin-Lei, Wang Guo-Yin. One error integrity indication codes and performance analysis. Computer Science, 2009, 36(6): 97-100(in Chinese)
(陈龙，方新蕾，王国胤. 系列单错完整性指示码及其性能分析. 计算机科学，2009，36(6)：97-100)

[11] Li Ji-Guo, Yu Chun-Wu et al. Mathematical Basis for Information Security. Wuhan: Wuhan University Press, 2006(in Chinese)

(李继国，余纯武等. 信息安全数学基础. 武汉：武汉大学出版社，2006)

[12] Rizzo L. Effective erasure codes for reliable computer communication protocols. ACM Computer Communication Review, 1997, 27(2): 24-36

[13] Tao Jun, Sha Ji-Chang, Wang Hui. Research on the analysis of the threshold scheme based on data dispersal coding storage. Journal of Chinese Computer Systems, 2008, 29(2): 353-356(in Chinese)
(陶钧，沙基昌，王晖. 基于数据分散编码存储的门限方案分析研究. 小型微型计算机系统，2008，29(2)：353-356)

[14] Merkle Ralph C. Protocols for public key cryptosystems// Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California, USA, 1980: 122-134

[15] Hou Fang-Yong, Wang Zhi-Ying, Liu Zhen, Memory integrity verifying based on Hash tree hot-window. Chinese Journal of Computers, 2004, 27(11): 1471-1479(in Chinese)
(侯方勇，王志英，刘真. 基于 Hash 树热点窗口的存储器完整性校验方法. 计算机学报，2004，27(11)：1471-1479)

CHEN Long, born in 1970, Ph. D., professor. His research interests include computer forensics, network security and intelligent information processing.

WANG Guo-Yin, born in 1970, Ph. D., professor, Ph. D. supervisor. His research interests include data mining, rough set, granular computing, knowledge technology, and intelligent information security.

Background

For digital evidence is inherently vulnerable, and it is easy to be modified while it is very difficult to discover the modifications, people doubt the integrity of digital evidence more and more. One important application of hash in computer forensics is to compute and record the hash value of a forensic target disk during imaging to assure the integrity of the cloned copy.

Preserving the integrity of evidence data in the cloned copy is a fundamental problem. Otherwise, either the defendant or the prosecutor can easily challenge the validity of it. However, simply taking one cryptographic hash for the whole copy is not appropriate due to the nature of hash function that even if one bit inside the whole copy has corrupted, the hash value of the "damaged" one will not be the same as the previously computed hash value, thus the integrity of the hard disk cannot be verified and the evidence becomes useless.

Fine-grained data integrity checking methods could isolate a portion of corrupted data segments and assure the integrity of other data at the same time, so as to mitigate the disaster effect on the data by some random errors or intentional forging modification.

The issue of traditional method, i. e. using n hash to check n data objects respectively, lies that the hash data themselves will occupy huge amounts of storage space at a fine-grained level. For the case of low error ratio of data objects, it is more efficient by crossing-hash checking than traditional method. This paper proposes a new crossing-hash integrity checking method based on Galois field uniform partition of multi-dimension structure. The method can indicate multiple errors accurately and efficiently, and provides a scalable scheme for different applications with several parameters.