张忠杰研究员主持“粮食储运国家工程研究中心绿色储粮科技新视野”特约专栏文章之四

DOI: 10.16210/j.cnki.1007-7561.2024.05.004
阎磊, 罗桃英, 段刚, 等. 储粮信息系统弱密码增强多重散列保护方法研究[J]. 粮油食品科技, 2024, 32(5): 26-34.
YAN L, LUO T Y, DUAN G, et al. Enhancement multiple-hashing protection method on weak password in grain storage information system[J]. Science and Technology of Cereals, Oils and Foods, 2024, 32(5): 26-34.

储粮信息系统弱密码增强多重散列保护方法研究

阎　磊[1,2]，罗桃英[1]，段　刚[3]，白春启[1,2]，李　亮[4]，尹正富[5]

（1. 河南工业大学 粮食和物资储备学院，河南 郑州 450001；
2. 粮食储运国家工程研究中心，河南 郑州 450001；
3. 河南豫信数智科技有限公司，河南 郑州 450001；
4. 陕西西瑞（集团）有限责任公司，陕西 西安 710003；
5. 中国储备粮管理集团有限公司，北京 100039）

摘　要：使用弱密码是计算机信息系统安全的重要漏洞，在粮食储藏信息化应用中亦是如此。单纯的哈希散列加密算法对弱密码存在明显局限性。结合粮库软件系统应用实际，本文提出以哈希散列算法为基础，混入盐值和引入高进制映射等进行增强处理，多次复用散列运算的弱密码保护方法。验证分析表明，方法增加了弱密码的加密值强度，阻断了字典攻击或彩虹表攻击，实现了对弱密码传递和存储的有效保护，且复杂度较低，程序实现难度低，运算耗时均值约 2.5 ms，对用户无感。方法用于粮食储藏信息化建设管理，可提供良好的安全保障。

关键词：粮食储藏；储粮；智慧粮库；弱密码；哈希散列
中图分类号：TS205　　**文献标识码：**A　　**文章编号：**1007-7561(2024)05-0026-09
网络首发时间：2024-08-29 14:40:05
网络首发地址：https://link.cnki.net/urlid/11.3863.TS.20240828.1732.019

Enhancement Multiple-hashing Protection Method on Weak Password in Grain Storage Information System

YAN Lei[1,2], LUO Tao-ying[1], DUAN Gang[3], BAI Chun-qi[1,2], LI Liang[4], YIN Zheng-fu[5]

(1. School of Food and Strategic Reserves, Henan University of Technology, Zhengzhou, Henan 450001, China; 2. National Engineering Research Center of Grain Storage and Logistics, Zhengzhou, Henan 450001, China; 3. Henan Yuxin Digital Intelligence Technology Co., Ltd., Zhengzhou, Henan 450001, China; 4. Shaanxi Xirui (Group) Co., Ltd., Xi’an, Shaanxi 710003, China; 5. China Grain Reserves Group Ltd., Company, Beijing 100039, China)

Abstract: The use of weak passwords is a significant vulnerability in computer information system security, and this issue is also present in the digital applications of grain storage information systems. Simple hashing

收稿日期：2024-05-11
基金项目：陕西省重点研发计划项目（2020ZDLNY05-09）；国家粮食和物资储备局标准制订项目（质检办便函[2019]25 号）
Supported by: Key Research and Development Project of Shaanxi Province (No. 2020ZDLNY05-09); National Food and Strategic Reserves Administration Standard- setting Project (No. [2019] 25)
作者简介：阎磊，男，1977 年出生，硕士，高级工程师，研究方向为粮食储藏信息技术，E-mail: yanlei@haut.edu.cn。本专栏背景及作者简介详见 PC7-9

algorithms have obvious limitations when dealing with weak passwords. Based on the actual application of grain storage software systems, this paper proposed a method to enhance weak password protection by using a hashing algorithm as the foundation, combined with the introduction of salt values and high-order reflections for strengthening, along with multiple rehashing operations. Validation and analysis showed that this method increased the hashing strength of weak passwords, and effectively prevented dictionary attacks or rainbow table attacks, which could achieve the effective protection for the transmission and storage of weak passwords. Additionally, it had low complexity, and was easy to implement. It also had an average computational latency of approximately 2.5 ms, which was imperceptible to users. This method is suitable for the construction and management of digital grain storage information systems, providing robust security protection.

Key words: grain storage; storage; smart granary; weak passwords; hashing

粮食安全是“国之大者”。悠悠万事，吃饭为大[1]。我国实行中央和地方分级粮食储备制度。粮食储备用于调节粮食供求、稳定粮食市场，以及应对重大自然灾害或者其他突发事件等情况[2]。国家对于储备粮油实施严格管理，要求承储企业应当对中央储备粮实行专仓储存、专人保管、专账记载，保证中央储备粮账账相符、账实相符、质量良好、储存安全[3]。做好储备粮管理工作，除了要求工作者有强烈的责任心，也依赖于储粮管理技术手段的应用。以计算机技术为代表的信息技术的兴起和发展，推动了各行各业的深度变革，当然也包括粮食储藏行业。我国近年来较大规模的粮食储藏信息化建设工作包括了国家粮食购销领域监管信息化建设[4]和粮食收储供应安全保障工程建设[5]，其中由国家发展改革委、国家粮食局、财政部联合组织实施的“粮安工程”建设规划，核心内容之一是仓储智能化升级改造，规划实施有力的推动了我国粮食储藏智能化和信息化建设水平的跃迁，促进了从中央到地方各级粮食仓储企业粮食仓储保管方式的提升和转变。

大规模的粮食储藏信息化建设，在为粮食仓储企业提质增效的同时，也暴露出企业在信息化深层应用方面的短板和不足。显著和突出的问题是信息化人才的缺乏、建成系统的管理应用水平偏低[6-8]。粮食仓储企业主要建设的信息化应用系统包括了粮情检测系统、一卡通出入库系统、安防监控系统、综合业务系统、虫情检测系统、气体检测系统、数量监测系统、智能通风系统、氮气气调杀虫系统、财务管理系统、办公 OA 系统和综合布线系统等。大多数企业在“粮安工程”仓储智能化升级改造之前，只具有粮情检测系统的建设和使用经验，对大规模信息化建设的关注主要集中在应用功能层面，较少关注系统间数据共享、数据安全防护等。在基层粮食仓储企业中，信息化系统的建设单位层次差异性较大，既有航天信息、浪潮集团等大型企业，也有贝博电子、叶威科技等行业企业，更多的是粮库所在地的本地公司，库内基础软硬件系统及网络基础设施维护依赖于建设和外协单位。在身份认证和密码保护方面，基本没有采用公开密钥密码体系，大部分信息化应用系统采用了 MD5、SHA-1 等单向散列函数方法保存密码，但仍存在以明码传递和保存密码的应用系统，诸如“123456”、“111111”，“admin”等不安全的弱密码也广泛使用，不难看出，其面临以下一些攻击的可能：

（1）弱密码虽然在存储时采用 MD5、SHA-1 等散列函数加密，但也较难避免由于算法单向映射密文而产生的字典攻击或彩虹表攻击[9]，一旦用户数据库泄露，此类弱密码对应密文几乎失去了加密意义；

（2）信息系统的前后端间采用明文传递密码，在用户登录过程中，可能在网络中被截获数据包，直接泄露用户密码。

基于此，以前述粮食仓储企业信息化管理水平和基础条件为背景，本文提出混入盐值和引入高进制映射等增强处理并多次多重散列的弱密码保护方法，以期构建安全的粮食储藏信息化软件系统。

1 问题分析

1.1 MD5 散列方法及弱密码局限性

人类社会对密码的保护由来已久，粮食储藏信息化应用系统常采用 MD5 算法来保持系统用

户密码。MD5（Message-digest algorithm 5）消息摘要算法产生于 1992 年[10]，算法的核心思想是将不定长的信息数据作为输入，通过运算得到 128 bit 的定长的消息摘要或“指纹”作为输出。对于输入数据，MD5 方法按照每 512 bit 来进行报文划分，将其作为一个分组，不足 512 bit 的尾部数据进行数据填充。由此，每一个 512 bit 的报文分组均由 16 个 32 bit 的字（words）构成。经过运算后，最终得到一个 128 bit 的散列值。MD5 算法具有运算单向性特征，即无法由输出结果逆运算得到输入数据。同时，不同的输入数据得到相同消息摘要输出数据的可能性极小。

当输入数据为长度较短的密码数据，由于 MD5 输出结果的定长唯一性，可以通过穷举法进行试算攻击。定义 $M=H(P)$，其中 P 为输入的密码，h 为计算的消息摘要值，穷举攻击时，逐一计算攻击密码 P' 的消息摘要值 M'，即 $M'=H(P')$，若存在 $M'=M$，则攻击密码 P' 即为用户密码。虽然穷尽攻击最差情况下耗时惊人，用每秒运算达 10 亿次的计算机也需要 1.07×10^{22} 年[11]。但是，当用户使用“123456”、“111111”，“admin”等不安全的弱密码时，则有限穷举攻击即可试算出消息摘要值，从而破译出密码。虽然在各级粮食储藏信息化建设过程中，不断要求使用大小写字母+字母+特殊字符的强密码，但显然弱密码的使用仍然是较难避免的问题。在针对 50 个国家的 4 TB 的密码数据进行分析后，人们发现无论是世界范围内还是在我国，“123456”都是最常使用的密码[12]，当然在粮食储藏信息系统中也同样如此。以某省级粮食仓储企业智能化信息系统为例，该企业现有信息系统 7 个，其中 5 个系统的用户密码采用了 MD5 加密算法进行存储。系统用户名主要是职工号和系统维护账户，共计 148 个。采用“123456”和“111111”作为默认密码的有 4 个系统，另有 1 个系统的默认密码与用户名相同。系统上线运行 30 天后未修改密码的账号有 64 个，占用户总数的 43.2%，运行一年后未修改密码的账号有 31 个，占用户总数的 20.9%，存在较大数据安全隐患。

“123456”对应的 MD5 消息摘要值“e10adc3949ba59abbe56e057f20f883e”（16 进制）已成为广为人知的公开数据，本文撰写时，在百度搜索引擎搜索“e10adc3949ba59abbe56e057f20f883e”关键词，得到了 1 420 000 个相关结果，并显著指向“MD5”和“123456”相关项，基本失去了密码保护的意义，如图 1 所示。由此可见，弱密码使用 MD5 散列方法具有明显的局限性，其密码安全基本上得不到 MD5 算法的保护。

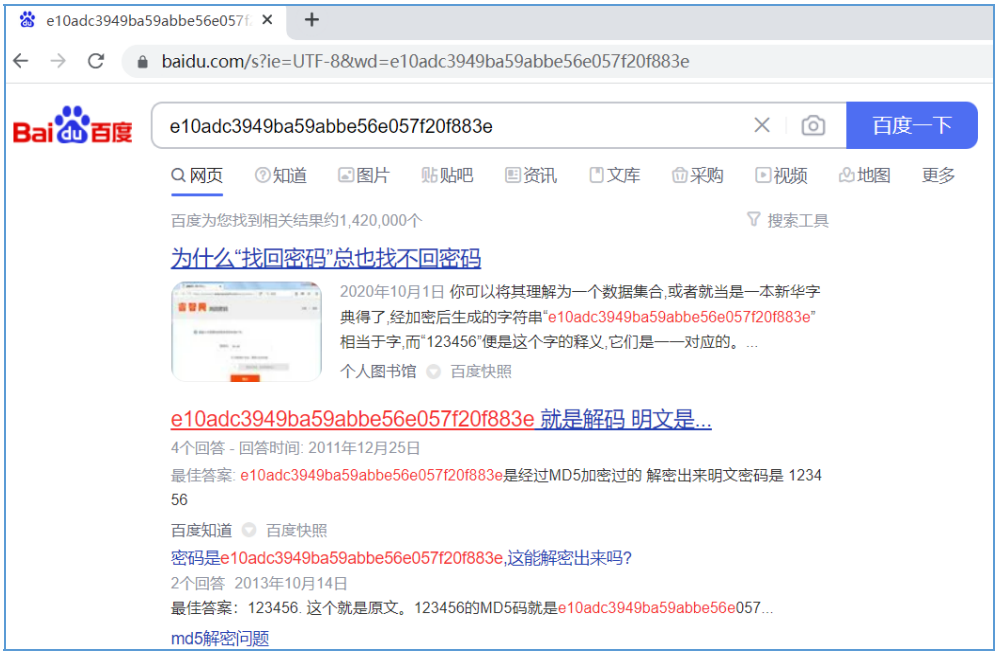

图 1 “e10adc3949ba59abbe56e057f20f883e”百度搜索引擎结果

Fig.1 Result of Baidu search engine for “e10adc3949ba59abbe56e057f20f883e”

1.2 B/S 形态密码明文传递隐患

基于 Web 的应用也已成为粮食储藏应用系统的主要形态，绝大多数软件采用了 B/S（Browser/Client）浏览器/客户端的软件体系结构。B/S 体系逻辑上一般划分为三个层次，包括接口层、逻辑层、数据层。也有的进一步细化为五层或七层逻辑架构。用户登录系统时，用户名和密码通过接口层进行输入，传递给逻辑层进行验证，验证的对象来自于数据层，即数据从前端接口层传递给后端逻辑层进行验证。目前的 Web 应用中，JavaScript 对象标记（JavaScript object notation，JSON）[13]数据交换格式标准也已成为事实上的前后端数据传递标准，大量系统采用 JSON 标准进行数据传递和交换。典型的用户登录 JSON 数据如表 1 所示。

表 1 用户登录 JSON 数据格式表

Table 1 JSON data format table of user login

```
{
    "username": "user1",
    "password": "password1"
}
```

JSON 是基于文本的数据格式，因此也被成为“JSON 字符串”。很容易从 JSON 数据中识别出其表示的含义。基于 JSON 的“字符串”特性进行用户登录数据传递时以明文形态显示，采用网络数据包捕捉工具对表 1 用户登录 JSON 数据进行捕捉并解析的结果如图 2 所示。

```
0200  3d 30 2e 39 0d 0a 0d 0a  7b 22 75 73 65 72 6e 61   =0.9···· {"userna
0210  6d 65 22 3a 22 75 73 65  72 31 22 2c 22 70 61 73   me":"use r1","pas
0220  73 77 6f 72 64 22 3a 22  70 61 73 73 77 6f 72 64   sword":" password
0230  31 22 7d                                          1"}
```

图 2 捕捉并解析用户登录 JSON 数据的结果

Fig.2 Capture and analyze the results of user login JSON data

2 弱密码增强多重散列方法

2.1 高进制数

人类社会通常采用十进制，也曾采用过五进制等[14-15]。计算机处理的是二进制，但也常用八进制、十六进制等来进行数的表示。十六进制引入了“A”、“B”、“C”、“D”、“E”、“F”五个字符来作为数字 0~9 的补充，构成十六进制的字符表现形式。同样道理，可以定义其他进制来满足数据处理的需要[16]，如三十四进制、六十二进制等。将 26 个英文小写字母“a”~“z”和 26 个英文大写字母“A”~“Z”作为数字 0~9 的补充，构成六十二进制。六十二进制可被用于压缩数据字节长度，也可以作为无符号的 Base64 进制编码使用[17]。十进制与六十二进制对应关系见表 2。

表 2 十进制与六十二进制对应关系表

Table 2 Conversion table of decimal to Base 62

十进制	六十二进制	十进制	六十二进制	十进制	六十二进制	十进制	六十二进制
0	0	16	g	32	w	48	M
1	1	17	h	33	x	49	N
2	2	18	i	34	y	50	O
3	3	19	j	35	z	51	P
4	4	20	k	36	A	52	Q
5	5	21	l	37	B	53	R
6	6	22	m	38	C	54	S
7	7	23	n	39	D	55	T
8	8	24	o	40	E	56	U
9	9	25	p	41	F	57	V
10	a	26	q	42	G	58	W
11	b	27	r	43	H	59	X
12	c	28	s	44	I	60	Y
13	d	29	t	45	J	61	Z
14	e	30	u	46	K		
15	f	31	v	47	L		

由表 2 可将十进制数对应转化为六十二进制数，例如十进制的“123456”转换为为六十二进制数计算过程如表 3 所示。即 $123456(10)=w7e(62)$

六十二进制数“$w7e$”转化为十进制数的计算过程为 $w\times62^2+7\times62^1+e\times62^0=123456$。同理，十进制的“123456”的 MD5 运算结果，十六进制

表 3 十进制数“123456”转换为六十二进制数计算过程

Table 3 Convert process form 123456(10) to w7e(62)

除数（基数）	被除数	商	余数	六十二进制数
62	123456	1991	14	e
62	1991	32	7	7
62	32	0	32	w

的“e10adc3949ba59abbe56e057f20f883e(16)”转化为六十二进制数为“6QEeFWWitRx7shDjF6RteC(62)”。

通过转化为六十二进制数的增强处理，可显著屏蔽原有值的可读性，同时也增加了原有值的复杂性，扩展了表达范围，相较于十六进制的字符串表示，六十二进制增加了至少 26 个表示字符。由于进制转换的可逆特点，在进行进制转化后，应再进行单线散列运算，已避免逆向运算获得原有值。

当然，六十二进制在计算机系统内仍属相对常见运算，仍可进一步提高进制，并采用乱序的进制对应关系，从而提高算法复杂性。在六十二进制基础上，再增加“~”、“!”、“@”、“#”、“$”、“%”、“^”、“&”等 8 个字符，变为七十进制，同时混乱若干进制对应关系，如 $3(10) \Leftrightarrow m(70)$，$21(10) \Leftrightarrow T(70)$，$60(10) \Leftrightarrow b(70)$，则可得到特殊且唯一的高进制映射关系。

2.2 增强处理的规则和定义

考虑弱密码传递隐患和存储的局限性，首先应避免密码的明文传递，对密码数据进行数据混入处理，并增强散列运算次数，从而隐藏密码本身。可通过混入随机值或特定关联值，增加原数据长度后，再进行多次多重散列的方法进行保护，混入的值可被称为“盐”（Salt）[18]，混入盐的方法可称为加盐。考虑弱密码数据传递和存储面临的不同风险，混入盐值后进行多重散列保护，定义规则如下。

（1）盐值应具有一定长度，混入原数据后可显著提高散列消息摘要运算结果不可知性。

（2）密码数据存储混入的盐不显性可见，即在数据库中不显然具有存储盐数据的字段。

（3）密码数据传递混入的盐应为非存储随机值，一次性使用，即完成数据传递验证后丢弃不保存。

（4）不同用户的相同密码数据经多重散列后得到的密文，保存在数据库中的值互不相等。

（5）混入盐计算方法应具有单项不可逆性，且易于实现。

（6）密码密文数据由特定高进制映射方法参与运算。

（7）散列方法可多次复运用以提高安全性。

基于此规则，给出如下定义。

定义盐：

$$S = H(Rand) \qquad \text{式（1）}$$

其中，S 是计算的盐值，H 为散列运算，$Rand$ 为足够长随机数（不少于 64 bit），生成方法可为多重素数算法、混沌算法、或梅森旋转算法[19-21]。

定义密码密文：

$$P_m = H(Base62(H(U) + H(P))) \qquad \text{式（2）}$$

其中，P_m 是密码多重散列值，H 为散列运算，U 为用户名字符串，P 为用户密码字符串，“+”运算是将字符串拼接运算，即将散列运算结果表示为 16 进制字符串形式，再进行拼接运算，运算结果仍为字符串。$Base62$ 为高进制（六十二进制）数据转换运算，运算结果为六十二进制字符串形式。由三次多重散列计算得到 P_m。

定义传递密文：

$$P_j = H(S + P_m) \qquad \text{式（3）}$$

其中，P_j 是粮食储藏应用系统前端向后端传递的密码多重散列值，H 为散列运算，S 是计算的盐值，P_m 是密码多重散列值。

定义验证密文计算方法同传递密文。

定义用户信息线性表：

$$Tuser = ((Uname_1, Pwd_1), (Uname_2, Pwd_2), \cdots\cdots (Uname_m, Pwd_m)) \qquad \text{式（4）}$$

用于表示数据库中存储用户名和密码多重散列值。其中，$Tuser$ 为二维线性表，表示用户信息，由一组含有两个数据项的数据域 $(Uname, Pwd)$ 组成。$Uname$ 表示的是用户名，即公式（2）中所述的 U；Pwd 表示的是用户 U 所对应的密码多重散列值，即的 P_m。

2.3 方法实现过程

以数据库表的形式定义用户信息的线性表 Table_UserInfo，存储于粮食储藏应用系统的数据层，其中主要包括了用户名和密码多重散列值，方法按公式（4）。用户名和密码多重散列值在用户创建时生成，密码多重散列值 P_m 与用户名 U 存在相关性，计算方法按公式（2）。

逻辑层是主要实务处理层，负责产生密码盐

和密文验证。当接口层激活粮食储藏应用系统的登录页面，逻辑层即生成盐值 S，并反馈给接口层。

接口层接收逻辑层传递的盐值 S，在输入登录用户名和密码后，由登录页面生成对应的传递密文 P_j，计算方法按公式（3），和用户名 U 一起提交传递给逻辑层。传递的 JSON 数据格式如表 4 所示。

表 4　用户登录多重散列密文 JSON 数据格式表

Table 4　Multiple-hashing JSON data format table of user login

```
{
    "username": " U ",
    "password": " P_j "
}
```

逻辑层接收接口层登录请求，提取用户名 U，带入盐值 S，向数据层查询对应的密码多重散列值，按公式（4）计算验证密文 P_v，若 $P_v = P_j$，则验证通过，用户身份得到确认。此时，反馈接口层验证成功消息，同时丢弃盐值 S，即盐值 S 为一次性非存储盐，即用即弃。

实现方法逻辑图如图 3 所示。

3　验证分析

3.1　实验结果

以粮食储藏信息化应用系统常用用户名密码数据进行实验，选取用户名“zhangwei”和密码“123456”。采用 MD5 散列方法，实验过程如图 4 所示。

经过运算，最终得到密码多重散列值“f30e1b145f662a71674fcf12bbd77c55 (16)”和传递密码“e7ff3d2bd7c194caffe7d4c1db835e6d(16)”，其中经过了 4 次多重散列运算。

3.2　安全性分析

根据实验结果可知，在引入高进制映射，增加非存储一次性盐值，并多次多重散列运算后，原有的不安全弱密码得到了较好的保护。特别是方法重复利用了散列运算的单向不可逆性。

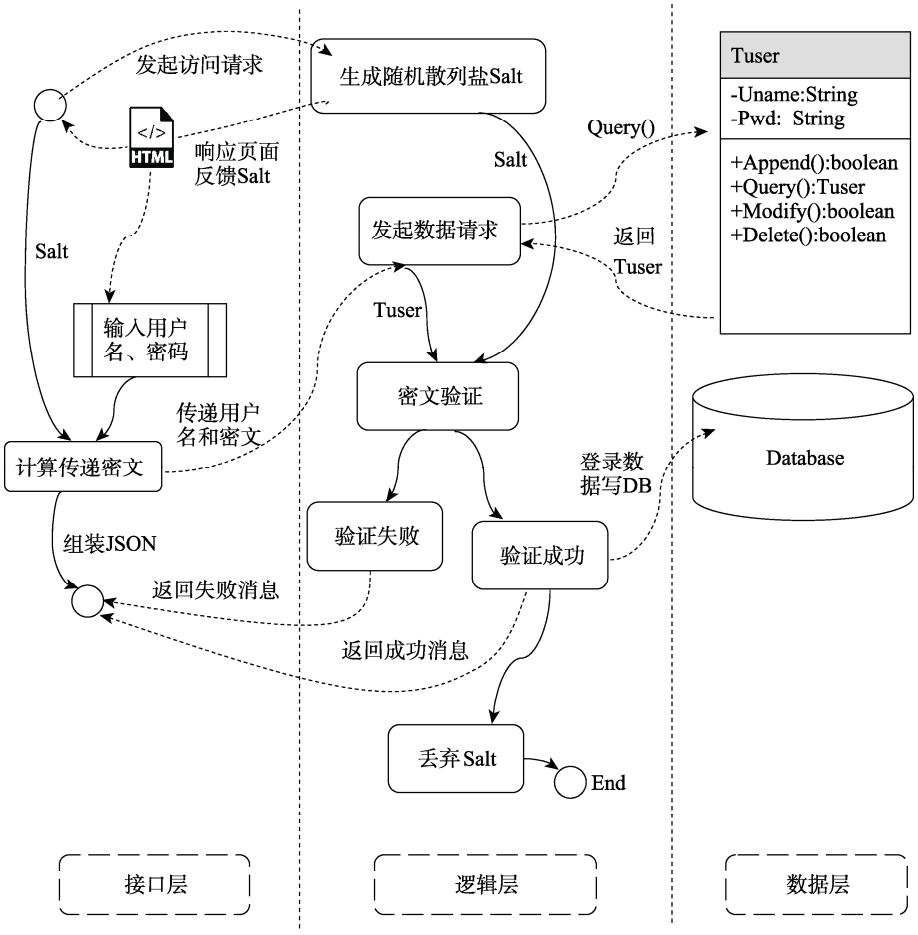

图 3　实现方法逻辑图

Fig.3　Logic diagram of implementation method

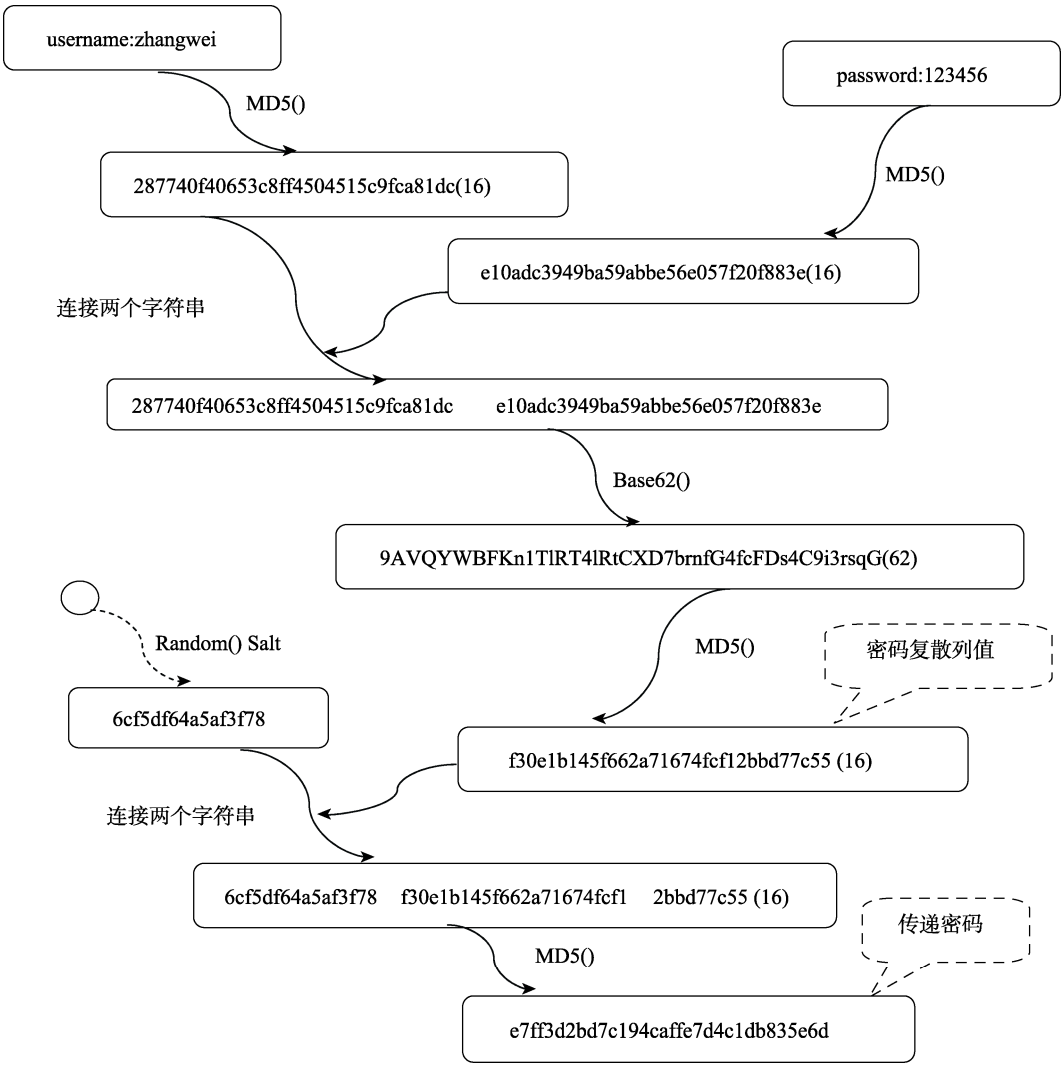

图 4 实验过程图

Fig.4 Experimental process diagram

（1）增加的非存储一次性盐值，随机产生，不在数据库中存储，即用即抛，与用户身份无相关性。

（2）高进制映射的引入，显著增加了加密变化过程中组合字符的熵值，其熵 $H \approx 77.07$，从而提高了后续处理密码的强度，且具有良好的熵保持性。

（3）前后端密码传递的保护，在密码传递中复用了 4 次散列运算，充分隐藏了原有密码信息。

（4）存储在数据库中的密码多重散列值计算中，加入了用户名信息作为干扰项，因系统中用户名的唯一性，使得相同的密码得到不同的散列值，进一步混淆了密码存储信息。

（5）常用哈希散列算法 MD5 和 SHA-1 算法存在数据碰撞的可能性[22-23]。但产生碰撞的数据并非能快速运算得到，寻找碰撞的计算量巨大，基本不具有可操作性，混入盐值和引入高进制映射后，算法具有良好的安全性。

3.3 性能分析

增强多重散列方法中主要算法耗时在于散列运算的多重循环和六十二进制数的转换，考虑接口层 Javascript 代码实现，对其进行 10 次耗时性能测试，测试服务器端硬件参数为处理器 Intel® Xeon® CPU E5-2640 2.40 GHz、内存 16 GB、硬盘 1 TB、网卡 Intel Gigabit-Ethernet-Controller，软件参数为操作系统 CentOS Linux release 7.8.2003 (Core)、数据库服务系统 mysql Ver 14.14 Distrib 5.7.33、Web 服务系统 Apache/2.2；测试客户端硬件参数处理器 Intel® Core™ i5-10500 CPU@3.10 GHz 3.10 GHz、内存 8 GB、硬盘 1 TB、网卡 Intel Ethernet-Controller1219，软件参数为操作系统 Windows 11

家庭中文版、浏览器 Chrome 125.0.6422.142。

测试结果如图 5 所示。

累计 10 次的耗时测试结果可以看出，最高耗时为 2.902 ms，最低耗时为 2.088 ms，耗时极差为 0.814 ms，耗时均值为 2.497 ms，运行性能较好，对用户无感。

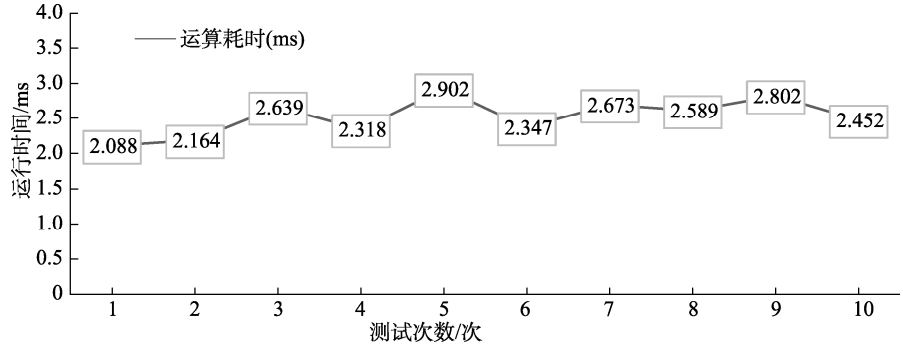

图 5 运算性能测试耗时

Fig.5 Time used for computing performance testing

4 结论

在粮食储藏信息化应用中，粮库软件的弱密码较难避免。单独使用哈希散列算法在弱密码保护中存在局限性。结合粮库应用实际，普遍建立和采用公钥体系尚难实现。基于此，本文以哈希散列算法为基础，混入干扰项和引入高进制映射进行增强，多次复用，大大增加了弱密码的加密值强度，阻断了字典攻击或彩虹表攻击，实现了对弱密码传递和存储的有效保护。相较于单纯的哈希散列加密，本文方法的算法显著提升安全性，且复杂度较低，不需要特殊程序处理，不影响软件功能，程序实现难度低，运算耗时较少，毫秒级处理对用户无感。

需要指出的是，虽然本文的方法提高了弱密码的保护效果，但并不认为在粮食储藏信息化应用系统中使用弱密码，粮库软件系统更应该提高安全性，采取技术手段限制和禁止弱密码的使用。安全的密码应该具有足够的长度，一般不少于 8 位字符，密码应避免采用全数字、全字符、连续符号、有明显意义的数字组合（如生日）或字符串等，可采用数字、字符、特殊符号随机组合的密码，如“KJi4~n!M6_9@i”，类似这样的密码可显著提高系统安全性。此外，粮食仓储企业和各级涉粮机构应构建信息安全保护体系，加强信息化管理和教育，提升人员信息化素养，才能从根本上保障信息化系统的安全有序运行。

参考文献：

[1] 张晓松，林晖. 习近平谈粮食安全：悠悠万事，吃饭为大[N]. 新华每日电讯, 2022-03-07(1).
ZHANG X S, LIN H. Xi Jinping talks about food security: Of all things, eating matters most[N]. Xinhua Daily Telegraph, 2022-03-07(1).

[2] 李玥. 深入贯彻《粮食流通管理条例》强化标准引领质量安全监管服务粮食产业高质量发展[J]. 粮食科技与经济, 2022, 47(S1): 1-3.
LI Y. Deeply implementing The Grain Circulation Management Regulation, strengthening standards, leading quality and safety supervision, and serving the high quality development of the grain industry[J]. Food Science And Technology And Economy, 2022, 47(S1): 1-3.

[3] 中央储备粮管理条例[J]. 中华人民共和国国务院公报, 2003(28): 13-18.
Regulations on the Administration of Central Grain Reserves[J]. Decree No. 388 of the State Council of the People’s Republic of China, 2003(28): 13-18.

[4] 张亨. 智慧粮库在粮食购销领域专项整治的应用探索[J]. 中国信息化, 2024(2): 73-74.
ZHANG H. Exploration of the application of smart grain depot in the special rectification of grain purchase and sales[J]. China’s Informatization, 2024(2): 73-74.

[5] 《粮食收储供应安全保障工程建设规划(2015-2020 年)》发布实施[J]. 粮油食品科技, 2015, 23(4): 116.
Implementation of “the Construction Plan for Grain Harvesting, Storage, and Supply Security Project” (2015-2020)[J]. Science and Technology of Cereals, Oils and Foods, 2015, 23(4): 116.

[6] 吴子丹. 新世纪中国粮食储藏科技发展新脉络的梳理与展望[J]. 粮油食品科技, 2023, 31(5): 9-18+276.
WU Z D. Review and prospect of China’s grain storage science and technology development in the new century[J]. Science and

Technology of Cereals, Oils and Foods, 2023, 31(5): 9-18+ 276.

[7] 戴鑫, 陈俊强, 赵利伟, 等. 信息化技术在粮食仓储管理中应用的探讨[J]. 粮食与食品工业, 2023, 30(4): 42-44.
DAI X, CHEN J Q, ZHAO L W, et al. Discussion on the application of information technology in grain storage management[J]. Cereal & Food Industry, 2023, 30(4): 42-44.

[8] 吴永刚, 张钊, 肖春燕, 等. 智能粮库建设过程中存在的问题与解决建议探析[J]. 粮食问题研究, 2022(5): 39-43.
WU Y G, ZHANG Z, XIAO C Y, et al. Analysis of the problems and solutions in the construction process of intelligent grain depots[J]. Grain Issues Research, 2022(5): 39-43.

[9] ABHILASH C, ANUPAM B, AJOY K K, SRIJAN. Secure randomized internally joined adjustable network for one-way hashing[J]. Journal of Information Security and Applications, 2024: 81.

[10] MOHAMMED A A, KADHIM F A. A novel improvement with an effective expansion to enhance the MD5 hash function for verification of a secure E-document[J]. IEEE Access, 2020, 8: 80290-80304.

[11] 张裔智, 赵毅, 汤小斌. MD5 算法研究[J]. 计算机科学, 2008(7): 295-297.
ZHANG Y Z, ZHAO Y, TANG X B. MD5 Algorithm[J]. Computer Science, 2008(7): 295-297.

[12] NordPass. Top 200 most common passwords[EB/OL]. [2022-05-06]. https://nordpass.com/most-common- passwords-list/.

[13] 罗武, 沈晴霓, 吴中海, 等. 浏览器同源策略安全研究综述[J]. 软件学报, 2021, 32(8): 2469-2504.
LUO W, SHEN Q N, WU Z H, et al. State-of-the-art survey of research on browser's same-origin policy security[J]. Journal of Software, 2021, 32(8): 2469-2504.

[14] 张斌荣, 曹少彰. 数学进制的发明与应用[J]. 发明与创新(大科技), 1999, (2): 18.
ZHANG B R, CAO S Z. The invention and application of mathematical base[J]. Invention and Innovation, 1999, (2): 18.

[15] 王坤鹏. 西周"亩臣"考[J]. 中国史研究, 2023, (1): 23-38.
WANG K P. Research on the "Muchen" in the Western Zhou Period[J]. Journal of Chinese Historical Studies, 2023, (1): 23-38.

[16] NAN H, JIANG J, ZHANG J, et al. Conversion between number systems in membrane computing[J]. Applied Sciences. 2023; 13(17): 9945.

[17] LIU Z, LU L, HILL R, et al. Base62x: An alternative approach to Base64 for non-alphanumeric characters[C]//Eighth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, 26-28 July 2011, Shanghai, China. IEEE, 2011.

[18] SUTRIMAN, SUGIANTORO B. Analysis of password and salt combination scheme to improve hash algorithm security[J]. The Science and Information (SAI) Organization Limited, 2019, 10(11): 420-425.

[19] OKADA K, ENDO K, YASUOKA K, et al. Learned pseudo-random number generator: WGAN-GP for generating statistically robust random numbers[J]. PLos One, 2023, 18(6).

[20] ZANG H, ZHAO X, WEI X. Construction and application of new high-order polynomial chaotic maps[J]. Nonlinear Dynamics, 2022, 107: 1247-1261.

[21] MAURO J D, SALAZAR E. SCOLNIK H D. Design and implementation of a novel cryptographically secure pseudorandom number generator[J]. Journal of Cryptographic Engineering, 2022, 12: 255-265.

[22] WANG X. How to break MD5 and other hash functions[J]. Eurocrypt, 2005.

[23] SADEGHI-NASAB A, RAFE V. A comprehensive review of the security flaws of hashing algorithms[J]. Journal of Computer Virology and Hacking Techniques, 2023, 19: 287-302. 完

备注：本文的彩色图表可从本刊官网（http//lyspkj.ijournal.cn）、中国知网、万方、维普、超星等数据库下载获取。