计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(10):80−85 [doi: 10.15888/j.cnki.csa.007108]

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

基于加盐 BCrypt 算法的电商安全模块设计及实现①

扈　玮, 王立华

(山东科技大学 电子信息工程学院, 青岛 266590)
通讯作者: 王立华, E-mail: 530243850@qq.com

摘　要: 随着大数据时代的来临, 密码泄露的情况时有发生, 数据安全已成为我们日益关心的问题. 本文运用 springmvc+spring+mybatis 框架技术, 通过模型分析、数据库表设计、时序图逻辑跳转的方式, 详细地阐述了加盐 BCrypt 算法在电商安全模块中的应用, 有效地解决了 MD5 加密算法的弊端, 极大地提高了信息的安全性.

关键词: MD5 加密算法; 加盐 BCrypt 算法; 密码加密; 密码验证

引用格式: 扈玮,王立华.基于加盐 BCrypt 算法的电商安全模块设计及实现.计算机系统应用,2019,28(10):80–85. http://www.c-s-a.org.cn/1003-3254/7108.html

Design and Implementation of E-Commerce Safety Module Based on Added Salt BCrypt Algorithm

HU Wei, WANG Li-Hua

(College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

Abstract: With the advent of the era of big data, the situation of password leakage occurs, data security has become an increasingly concerned issue. Using springmvc+spring+mybatis framework technology, this paper expounds in detail the application of added salt BCrypt algorithm in e-commerce security module by means of model analysis, database table design, and timing diagram logic jump, which effectively solves the drawbacks of MD5 encryption algorithm, and greatly improves the security of information.

Key words: MD5 encryption algorithm; added salt BCrypt algorithm; password encryption; password authentication

电子商务的快速发展对系统的安全性提出了更高的要求, 但是许多企业应用的安全性还没有得到足够的重视, 密码泄露的情况时有发生, 造成了严重的经济损失和社会不安. 早在 2008 年, 卡内基梅隆大学软件工程研究所就通过美国政府资助的漏洞警报系统向安全专业人员发出公告: MD5 应被视为已破解且不适合继续使用的加密方法; 网络公司 TrustedSec LLC 的首席执行官 David Kennedy 指出: “MD5 被认为在 2013 年之前就已经过时了. 从那时起, 大多数公司开始使用更安全的哈希算法”[1]. 在 2019 年 01 月, BMG 公司正式确认该公司的热门游戏《塞勒姆小镇》有安全漏洞, 该漏洞现已得到解决, 但该游戏拥有 760 万用户, 其信息受到损害. 用户的密码、IP 地址和电子邮件信息有被泄露的风险, 而泄密的原因是: 使用 MD5 哈希算法存储帐户密码[2]. 随着此类密码泄露事件的一次次发生, 迫切需要更安全的加密算法提高系统的安全性能.

运用 MD5 加密算法虽然可以实现密码加密的功能, 产生 32 位字符串并且算法不可逆. 但是 MD5 加密

① 基金项目: 山东省自然科学基金 (ZR2018MF005)
Foundation item: Natural Science Foundation of Shandong Province (ZR2018MF005)
收稿时间: 2019-03-18; 修改时间: 2019-04-17; 采用时间: 2019-04-23; csa 在线出版时间: 2019-10-15

算法存在着诸多弊端: 同一密码经过 MD5 算法产生的字符串一模一样; 同时随着计算机技术的发展和计算水平的不断提高, 可以用彩虹表等方法轻松破解, 已不再适合安全要求较高的场合使用[3–6]. 针对此类情况作出改进, 旨在设计并实现一个以 springmvc+spring+mybatis 框架为基础的电商安全模块[7–11], 其中表现层 springmvc 框架用于客户端的视图显示; 业务层 spring 框架用于系统业务逻辑操作的具体实现; 持久化层框架 mybatis 用于与 Mysql 数据库进行数据交互, 完成增删改查 (CRUD) 操作. 同时用更安全的加盐 Bcrypt 算法实现密码加密和验证功能, 加强了数据安全的维护工作.

1 加盐 BCrypt 算法

1.1 BCrypt 算法

Bcrypt 算法是基于 Blowfish 加密设计的一个慢哈希算法, 是 Open BSD 或其他 Linux 发行版的默认加密算法, 该算法的设计是为了抵抗对哈希值或者盐的布鲁特攻击, 有效地解决了 MD5 加密算法的弊端[12–17].

算法原形是: *DK*= *bcrypt*(*cost*, *salt*, *input*)

参数 *cost* 是计算强度, 即循环次数;

参数 *salt* 是密码学中的盐;

参数 *input* 是输入的密码文本;

输出 *DK* 是生成的派生密钥.

BCrypt 算法计算后的密文中包含了算法版本、循环次数、盐值以及哈希字符串. 下面是“123456”以 10 为循环次数经加盐 BCrypt 算法加密后随机生成的 60 位字符串“$2a$10$jI49TG.oPta0fuhlxqtLzeUZuwN1rzZU9G/1R1CIYIT.lirNPimwu”. 其中$是分割符, 无意义; 2a 表示 BCrypt 算法加密版本号; 10 是 *cost* 值, 表示密码加密的计算强度, 强度越高, 则密码越复杂, 计算时间也越长; 而后的前 22 位是盐值, 具有随机性; 其余的哈希字符串就是密码的密文了.

1.2 加盐处理

如果只是单纯的哈希字符串存储, 如果两个或更多人设置的密码相同, 则加密后将得到相同的结果, 进入到数据库破解一个就可以破解一大片的密码. 例如名为 A 的用户可以查看数据库, 可以发现存储的哈希字符串结果有些是相同的, 那么不同的用户对应的就是同一个设置密码, 这样就可以利用别人的身份登录系统. 由于盐是随机产生的, 存储不同用户数据时加入的盐都是随机的, 所以就可以保证不同的用户即使有相同的密码, 存储在数据库中的密码也是千差万别的. 上述加密策略, 可以形象地称为“加盐哈希加密”[18], 可以在对密码进行哈希算法之后, 将盐随机并混入加密后的哈希字符串, 进行二次加密, 验证时也无需单独提供之前的盐, 从而无需单独处理盐的问题, 更加安全方便.

BCrypt 是单向的, 而且经过 *salt* 和 *cost* 的处理, 与 MD5 加密算法相比, 安全性得到了极大地提高. 为使攻击者难以构造出所有可能值的查询表, 其盐值和计算强度的值必须足够长, 彩虹表攻击破解密文的概率则越低, 破解的难度也就越大.

2 加盐 BCrypt 算法的模型分析

在电商安全模块中, 用户密码的加密和验证流程如图 1 所示. 流程包括用户注册和用户登录两大部分[19]. 有信息系统存在的地方, 就需要进行用户信息认证, 还可以根据用户赋予不同的访问权限. 安全模块的注册和登录的流程大致如下:

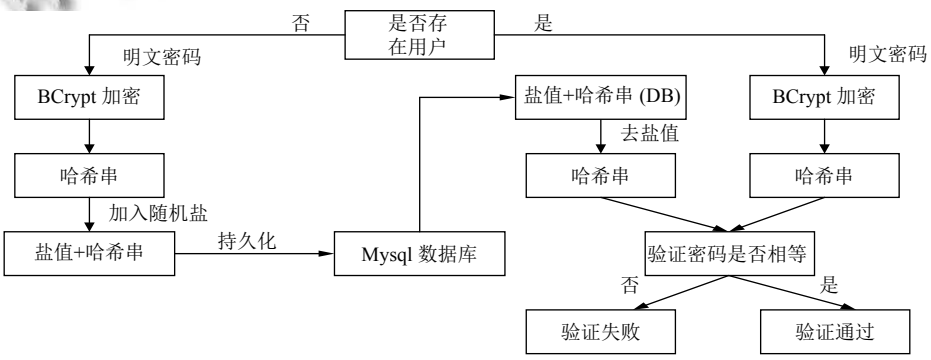

图 1 密码加密存储与验证流程图

(1) 用户注册: 用户输入登录名、密码及其他有关信息后, 将相关数据传输到后端服务器, 直至保存到 Mysql 数据库中. 因为登录名也是主键, 具有唯一性, 若输入的登录名已经存储在数据库中, 则无法成功注册, 仍就停留在注册页面并且弹出注册失败的提示;

(2) 后端服务器接受到用户数据, 将其中的密码进行加盐 BCrypt 算法加密处理. 若主键不重复, 则表单数据和对应的创建日期存储到数据库中, 页面跳转到登录页面;

(3) 用户登录: 用户输入的登录名及密码传输到服务器, 服务器通过登录名从 mysql 数据库中查询该用户是否存在, 若用户不存在, 则仍就停留在登录页面;

(4) 若用户存在, 则服务器将客户端传过来的密码经过 BCrypt 算法加密处理得到哈希字符串; 同时服务器根据登录名取得数据库中对应的密文, 并提取出盐值得到哈希字符串; 并对两个哈希字符串进行验证, 如果完全相同, 则表示用户登录成功, 页面跳转到用户主页; 如果不同, 则表明用户登录失败, 仍就停留在登录页面, 并且不会给出任何信息提示.

3 加盐 Bcrypt 算法的实现

用户注册和登录功能涉及到用户表 tb_user, 如表 1 所示; 涉及到的类, 如表 2 表示.

3.1 用户注册的密码加密功能

用户注册的时序图如图 2 所示, 时序图中涉及到的主要方法介绍:

Add (@RequestBody TbUser user): 后端控制层增加用户信息的方法;

encode (rawPassword): 加密方法, 参数 rawPass- -word 为用户输入的密码;

add (TbUser user): 后端服务层增加用户信息的方法;

insert (user): 后端数据访问层向数据库增加用户信息的方法.

表 1 用户表 (tb_user)

字段	类型	长度	含义
user_id	varchar	100	主键; 用户登录名
paeeword	varchar	60	密码
company_name	varchar	80	公司名称
nick_name	varchar	50	用户昵称
telephone	varchar	50	用户电话
address_detail	varchar	100	详细地址
create_time	datetime		创建申请日期

表 2 类的简介

类	简介
Result	自定义结果实体类, 包括 Boolean 类型的 success 字段, String 类型的 message 信息字段
UserController	后端控制类, 控制用户相关操作的业务逻辑和跳转关系
UserService	用户类别的服务接口类, 定义了对用户数据的操作
UserServiceImpl	用户类别的的服务实现类
TbUserMapper	TbUser 类的数据访问类, 用来操作数据库 tb_user 表
TbUser	用户实体类, 表 1 已给出具体字段
PasswordEncoder	密码工具接口类, 定义了加密方法和密码验证方法
BCryptPasswordEncoder	密码工具接口的实现类
UserDetailsServiceImpl	自定义认证类, 用于验证用户名、密码和权限角色

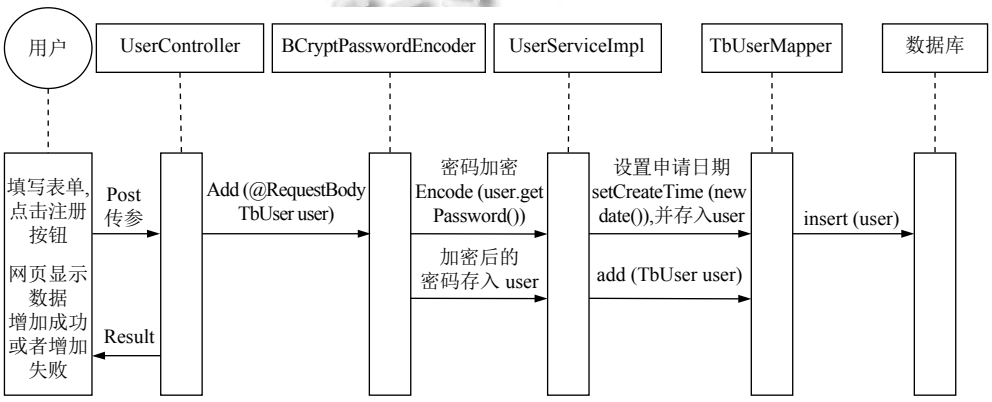

图 2 密码加密功能时序图

当用户在注册页面上填写表单, 点击注册按钮后, 将依次调用控制类 UserController 的 add (@Requ- -estBody TbUser user) 方法; BCryptPasswordEncoder 的 encode (rawPassword) 方法, 并将加密后的字符串存入

user 中; 服务接口类 UserService 的 add (TbUser user) 方法; 服务实现类 UserServiceImpl 的 add (TbUer user) 方法, 并设置申请日期存入 user 中; 数据访问类 TbUser Mapper 的 insert (user) 方法, 向数据库插入表单数据.

在控制类 UserController 中, 如果增加数据成功, Result 传递的参数为{success: true, message: “注册成功”}, 页面自动跳转到登录页面; 如果增加数据失败, Result 传递的参数为{success: false, message: “注册失败”}, 页面仍就停留在注册页面.

用户注册功能的实现效果图, 如图 3 所示.

图 3 注册功能实现图

3.2 用户登录的密码验证功能

用户登录的时序图如图 4 所示, 时序图中涉及到的主要方法介绍:

loadUserByUsername (String username): 认证类的验证方法, 参数为用户输入的登录名;

findOne (String id): 后端服务层根据 id 查询用户数据的方法;

selectByPrimaryKey (id): 后端控制层根据 id 从数据库查询用户数据的方法;

matches (rawPassword, encodedPassword): 密码验证方法, 参数 encodedPassword 表示从数据库读取的密文.

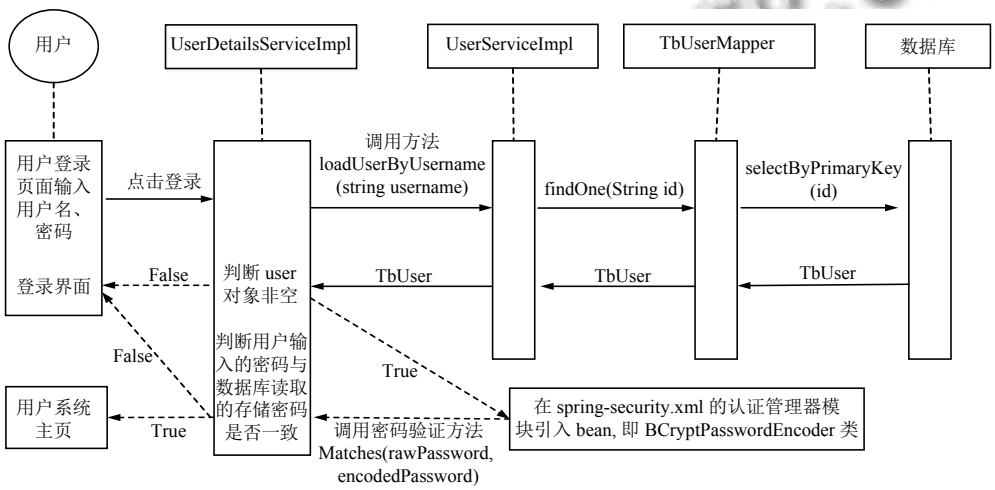

图 4 密码验证功能时序图

用户在登录页面输入登录名和密码, 点击登录按钮之后, 依次调用 UserDetailsServiceImpl 的 load- -UserByUsername(String username) 方法; UserService- -Impl 的 findOne(String id) 方法; TbUserMapper 的 selectByPrimaryKey(id) 方法, 运用 mybatis 框架的 xml 配置文件可以方便地从数据库中得到主键对应的

TbUser 用户对象, TbUserMapper.xml 文件里对应的查询语句如下:

<select id= "selectByPrimaryKey" resultMap= "BaseResultMap" parameterType= "java.lang.String">

select <include refid= "Base_Column_List"/> from tb_user

where user_id = #{ userId, jdbcType= VARCHAR}

</select>

其中 parameterType 表示输入参数是字符串类型, user_id 表示用户输入的登录名, Base_Column_List 表示主键对应的包括诸多字段信息的 TbUser 用户对象, resultType 表示把查询结果封装到 pojo 类型中.

在 UserDetailsServiceImpl 认证类中, 判断 user 对象非空, 若不满足条件则仍是登录页面, 若满足条件则在 spring-security.xml(安全框架配置页) 的认证管理器模块引入密码工具接口的实现类 BCryptPasswordEncoder, 调用 matches(rawPassword, encodedPassword) 方法进行密码验证, 判断用户输入的密码和从数据库读取的密码是否一致, 若不满足条件则仍是登录页面, 若满足条件则跳转到用户系统主页. 用户登录功能实现效果图, 如图 5 所示.

4 加密算法的实现效果

如图 6 所示, 数据库存储的密码均为 123456, xiaoA 用户的密 xiaoB 用户和 xiaoC 用户的密码都采用 MD5 加密算法, 产生了相同的 32 位字符串, 存储到数据库中. 管理员虽然不能直观看到账号的密码, 但是同样的明文经过 MD5 算法加密产生相同的密文, 而且可以用彩虹表等方法轻松破解. 采用加盐 BCrypt 算法成功解决了以上问题, 正如图中的 xiaoD 用户和 xiaoE 用户的密码都采用加盐 BCrypt 算法, 产生了随机的 60 位字符串, 存储到数据库中, 大大提高了数据的安全性.

图 5 登录功能实现图

user_id	password	company_name	nick_name	telephone	address_detail	create_time
xiaoA	123456	小A公司	小A	4004004400	东一路	2019-02-28 10:30:08
xiaoB	e10adc3949ba59abbe56e057f20f883e	小B公司	小B	010-0101010	东二路	2019-02-28 11:02:20
xiaoC	e10adc3949ba59abbe56e057f20f883e	小C公司	小C	010-0101010	东三路	2019-02-28 11:04:11
xiaoD	$2a$10$jI49TG.oPta0fuhlxqtLzeUZuwN1rzZU9G/1R1CIYIT.lirNPimwu	小D公司	小D	8008008800	东四路	2019-02-28 11:06:53
xiaoE	$2a$10$VNVyU0qN93VNADsuUm.3Qu/HALJYxOiYTH3UxLBU9f/VIGtBtRU1y	小E公司	小E	000-123456	东五路	2019-03-01 10:48:44
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

图 6 实现效果图

5 总结

现今信息安全形势不容乐观, 很多企业应用安全性未得到足够重视, 涉及购物的电商企业更应加强信息安全, 避免造成不可估量的损失. 本文结合高内聚低耦合法则、SSM 框架三层架构的思想, 详细阐述了用户注册和登录功能设计与实现过程, 采用加盐 BCrypt 加密算法实现密码存储加密和验证功能, 解决 MD5 加密算法的弊端, 为电商安全提供了一份有效解决方案.

参考文献

1 王凤枝. 被 MD5 加密坑了? 雅虎大规模泄密是个咋样的悲剧. http://tech.163.com/16/1219/09/C8KVR7PT00097U7R.html. [2016-12-19].

2 黑客视界. 760 万“塞勒姆小镇”玩家个人信息惨遭泄露. http://dy.163.com/v2/article/detail/E4RGVJ1H0511QOAF.html. [2019-01-06].
3 李夏梦, 潘广贞. 基于消息摘要算法第五版和 IDEA 的混合加密算法. 科学技术与工程, 2017, 17(9): 233–238. [doi: 10.3969/j.issn.1671-1815.2017.09.040]
4 罗江华. 基于 MD5 与 Base64 的混合加密算法. 计算机应用, 2012, 32(S1): 47–49.
5 Zhong LX, Wan WG, Kong DK. Javaweb login authentication based on improved MD5 algorithm. Proceedings of 2016 International Conference on Audio, Language and Image Processing. Shanghai, China. 2016. 131–135.
6 李夏梦. IDEA 子密钥扩展算法及其与 MD5 混合加密算法的研究[硕士学位论文]. 太原: 中北大学, 2017.
7 苏庭波, 王世权. 基于 SSM 的品优购后台管理系统的设计与实现. 江西科学, 2018, 36(5): 866–870.
8 陈峰. 基于 SSM 框架的 B2C 网上商城系统的设计与实现[硕士学位论文]. 长沙: 湖南大学, 2018.
9 朱重佳. 基于 SSM 框架的网购商城的设计与实现[硕士学位论文]. 北京: 北京交通大学, 2018.
10 周星宇. 跨境电商在线商城订单子系统的设计与实现[硕士学位论文]. 南京: 南京大学, 2018.
11 李洋. SSM 框架在 Web 应用开发中的设计与实现. 计算机技术与发展, 2016, 26(12): 190–194.
12 尚华益, 姚国祥, 官全龙. 基于 Blowfish 和 MD5 的混合加密方案. 计算机应用研究, 2010, 27(1): 231–233. [doi: 10.3969/j.issn.1001-3695.2010.01.068]
13 张勇, 张德运, 蒋旭宪. 基于认证的网络权限管理技术. 计算机工程与设计, 2001, 22(2): 52–55. [doi: 10.3969/j.issn.1000-7024.2001.02.014]
14 郭玉伟. 基于 SAML2. 0 的企业用户认证授权集中管理平台的设计与实现[硕士学位论文]. 广州: 华南理工大学, 2014.
15 张文超, 李贺, 程刚. 基于单向加盐慢哈希算法的密码安全存储的研究与实践. 中国数字医学, 2018, 13(5): 8–11. [doi: 10.3969/j.issn.1673-7571.2018.05.003]
16 Aggarwal A, Chaphekar P, Mandrekar R. Cryptanalysis of bcrypt and SHA-512 using distributed processing over the cloud. International Journal of Computer Applications, 2015, 128(16): 13–16. [doi: 10.5120/ijca2015906744]
17 Malvoni K, Designer S, Knezovic J. Are your passwords safe: Energy-efficient bcrypt cracking with low-cost parallel hardware. Proceedings of the 8th USENIX Conference on Offensive Technologies. San Diego, CA, USA. 2014. 10.
18 祝彦斌, 王春玲. 一种 Hash 特征隐藏的加盐信息摘要模型. 计算机技术与发展, 2013, 23(3): 134–138.
19 周小红, 周建伙. MD5 加密算法在注册及登录验证模块中的应用. 工业控制计算机, 2015, 28(11): 118–119. [doi: 10.3969/j.issn.1001-182X.2015.11.051]