3

安全的密码设计

密码设计的问题

- 密码可以明文存储吗?
- 对密码做对称加密可以吗?
- 对密码做不对称哈希可以吗?
- 还需要做什么呢?

密码保护环节

01 信源：用户输入密码

02 信道：密码传输过程

03 信宿：密码相关的存储

明文存密码是肯定不行的

thisisapassword

哈希函数能解决问题吗?

thisisapassword → 93f210ca9c640af9

好的哈希函数的特征

01 快速进行哈希，给出结果

02 不同的内容得到的哈希结果尽可能不同（几乎不碰撞）

03 微小的改动应该带来巨大的结果差异

04 难以逆向推得（往往伴随信息丢失）

弱哈希是肯定不行的

MD5(16b): 93f210ca9c640af9

MD5(32b): 15c4683193f210ca9c640af9241e8c18

SHA1: 1fa46b674402a47b2d32ef07481ccc7e251c5122

弱哈希是肯定不行的

MD5(16b): 93f210ca9c640af9

MD5(32b): 15c46831193f210ca9c640af9241e8c18

SHA1: 1fa46b674402a47b2d32ef07481ccc7e251c5122

都已被王小云等组成的团队
使用杂凑算法攻破

暂时还可用

01 SHA-256

02 SHA-384

03 SHA-512

04 SHA-224

05 Ripemd-160

但是用户很懒

1. 123456
2. Password
3. 12345678
4. qwerty
5. 12345
6. 123456789
7. letmein
8. 1234567
9. football
10. iloveyou

彩虹表

1.	123456	8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92
2.	Password	e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e19398b23bd38ec221a
3.	12345678	ef797c8118f02dfb649607dd5d3f8c7623048c9c063d532cc95c5ed7a898a64f
4.	qwerty	65e84be33532fb784c48129675f9eff3a682b27168c0ea744b2cf58ee02337c5
5.	12345	5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c173cacfc5
6.	123456789	15e2b0d3c33891ebb0f1ef609ec419420c20e320ce94c65fbc8c3312448eb225
7.	letmein	1c8bfe8f801d79745c4631d09fff36c82aa37fc4cce4fc946683d7b336b63032
8.	1234567	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f01665370476383ea776df414
9.	football	6382deaf1f5dc6e792b76db4a4a7bf2ba468884e000b25e7928e621e27fb23cb
10.	iloveyou	e4ad93ca07acb8d908a3aa41e920ea4f4ef4f26e7f86cf8291c5db289780a5ae

安全策略

01 不许用户设置弱密码

02 让用户使用手机验证等方式代替

03 采用生物特征代替密码

04 加盐（salting）

05 增加迭代哈希次数

加盐怎么加

01 代码中的静态盐

02 每个用户自己的动态盐

03 把盐存到和哈希结果不同的位置

04 更复杂的加盐策略

Django 默认怎么做

- 数据库存的是 <algorithm>$<iterations>$<salt>$<hash>
- 算法默认的是 PBKDF2 和 SHA256 的结合
- 可以有其他替换方案（见官方文档）