使用一种不可逆的加密算法，将要保存密码或者私钥进行加密处理。

一旦生成密码的 Hash 值并存储在数据库中后，你将不可能再把它转换回密码明文。只能每次用户在登录到应用程序时，须重新生成 Hash 值与数据库中的 Hash 值匹配来完成密码的校验。

一、直接明文密码

非常不安全

二、MD5/SHA的hash算法

- 最常见
- 可以字典破解或暴力破解，但复杂度大
- 可以用高效的查表法破解--逆向查询表及彩虹表法（rainbow table）

三、salt+hash算法

- salt（盐）为一个随机生成的长字符串，salt+原始密码 进行hash算法加密，即HASH(key+salt)
- 使用CSPRNG生成盐，而不是普通的随机数算法
- salt不能太短，且不能重复使用 (部分系统使用用户的字段、id、手机号作为salt)

四、对经过一次hash处理的密码加入salt进行再一次hash处理

- 密码先经过hash处理，再加入salt，再次进行hash处理，增加破解难度。

五、PBKDF2算法

- 常用的三种密码Hash算法之一
- salt+密码 多次进行hash运算，随机salt和多次HASH也使得建表和破解的难度都大幅增加。（密码学中称为拉伸）
- HASH算法一般选用sha1或者sha256，随机盐的长度一般不能少于8字节，HASH次数至少也要1000次
- 美国国家标准与技术研究院推荐使用的算法

六、bcrypt算法

- 常用的三种密码Hash算法之一。
- 基于Blowfish加密算法改进得到，只用于密码存储。
- 通过设定一个参数work factor调整计算强度，同时work factor是包括在输出的摘要中的。随着攻击者计算能力的提高，使用者可以逐步增大work factor，而且不会影响已有用户的登陆，
- 一般认为它比PBKDF2更能承受随着计算能力加强而带来的风险。
- bcrypt有广泛的函数库支持，建议使用这种方式存储密码。

七、scrypt算法

- 常用的三种密码Hash算法之一。
- 安全性应高于PBKDF2和bcrypt。
- scrypt算法所需计算时间长，而且占用的内存也多，使得并行计算多个摘要异常困难，因此利用rainbow table进行暴力攻击更加困难。
- 但是相比上述算法，scrypt没有在生产环境中大规模应用，并且缺乏仔细的审察和广泛的函数库支持。在memory hard hash中，scrypt 作为使用的拉伸算法，有效的计算每个内存访问是取决于前一个实例。

八、Argon2算法

- 2015 年 Password Hashing Competition 的胜出者，被NIST认定为最好的密码hash算法，目前没有被大量使用。
- 论文地址： https://password-hashing.net/argon2-specs.pdf
  github： https://github.com/khovratovich/Argon2
  https://github.com/P-H-C/phc-winner-argon2
- Nano使用此算法加密
- 目前认为该算法在安全性和成本效益方面优于当今使用最广泛的密码散列函数 Bcrypt。除了密码散列函数外，该算法还适用于与现代电子（加密）货币一起使用的工作证明操作。
- 可以设置时间成本（迭代次数），还能设置空间成本（内存占用），使得算法重度依赖内存，这样对于一些计算力很强但存储性能一般的设备（例如 GPU、ASIC 等）破解优势就大幅降低了。
- 支持多线程计算，因此同样的时间里可投入更多的工作量用于 Hash 计算，这让破解成本也增加数倍。

- 目前其他算法存在的问题：泛型模式PBKDF2、bcript和scrypt。其中，只有scrypt需要高内存，但存在一个无关紧要的time-memory允许紧凑的实现具有相同的成本。另一个问题就是复杂性很难进行分析。在时间成本和内存成本上不适用。
- Argon2结合了硬件内存设计中的最新技术。它旨在提供最高的内存填充率和多个计算单元的有效使用，同时还能防御tradeoff攻击。 Argon2针对x86架构进行了优化，并利用了最近英特尔和AMD处理器的缓存和内存组织。
- Argon2有两种变体：Argon2d和Argon2i。 两种版本的Argon2都可以在几秒内完成1 GB的RAM，而更小的数量甚至更快。它可以轻松扩展到任意数量的并行计算单元。其设计也进行了优化，以便于分析和实施。
- Argon2d速度更快，并且使用依赖数据的内存访问，这使得它适合加密货币和应用程序，并且不会受到来自侧通道定时攻击的威胁。 Argon2i使用与数据无关的内存访问，这是密码散列和基于密码的密钥派生的首选方式。Argon2i速度较慢，因为它可以在内存上传递更多信息，以防止受到tradeoff攻击。