1. Explain what are Secure Multi Party Computations and give a detailed example of their possible application. Discuss possible benefits of, and potential issues with your proposed application.

(20marks)

问题一：什么是安全多方计算

安全多方计算（ Secure Multi-Party Computation，MPC）于1986 年由姚期智院士提出【2】。安全多方计算协议允许多个数据所有者在互不信任的情况下进行协同计算，输出计算结果，并保证任何一方均无法得到除应得的计算结果之外的其他任何信息。换句话说，MPC技术可以获取数据使用价值，却不泄露原始数据内容。

问题二、可能应用的详细示例

我们首先介绍一种基础的安全多方计算协议：不经意传输(Oblivious Transfer, OT)。

来看一个例子：假设某旅行社拥有N个景点的旅游资料，小淘想去其中的A景点游玩，希望向旅行社购买相关资料做好出游功课。但是小淘非常在意自己的隐私，不希望向旅行社泄露自己的目的地是哪里。因此双方希望这笔交易能够满足以下隐私条件：

小淘不希望向旅行社泄露“我准备去A景点”这一信息；

旅行社只希望出售小淘出钱购买的那份资料，而不泄露小淘未购买的N-1份资料；

粗看起来这种隐私条件似乎是无法满足的：旅行社只要把景点A的资料给到小淘，就必然了解了“小淘正在关注A景点”这一信息；除非旅行社把所有N份资料都给出，但是这又违背了旅行社的利益；

但是神奇的OT可以让交易在这种“不可能的条件”下达成。简而言之，在OT协议中，旅行社把他拥有的N份资料使用某种双方协商同意的加密算法和参数进行加密，然后发送给小淘；小淘可以从密文中解密出A的资料，而无法解密出其他N-1份资料。

以下以N=2为例，基于Diffie-Hellman密钥交换协议，给出一种1 of 2 OT实现方法的非正式描述；其中S（Sender）=旅行社，R（Receiver）=小淘，S拥有两份资料image，R希望取得其中的image；

S秘密生成随机数a; R秘密生成随机数b；

S将image发送给R; R将image发送给S；

S计算image；

S以image为密钥加密image, 以k1为密钥加密image,将image和image发送给R；

由于image, 因此R可以计算出image，并解密出image，但R无法计算image，因此无法解密出image。

如果R希望取得image，只需把第2步中的image改为image即可。



OT除了可以直接用于构造MPC方案之外，也是GC等许多MPC方案的基石。

**参考文档：安全多方计算新突破！阿里首次实现“公开可验证” 的安全方案**

**https://yq.aliyun.com/articles/693224?spm=a2c4e.11153940.0.0.1141434coXQEhN**

2. The following algorithm has been proposed to compute a hash function.

*Generate some key K for the AES algorithm. Encrypt M with K and take the last 64 bits of the result as the hash of M.*

What are the main disadvantages of this algorithm? Can the algorithm be improved by taking the first 64 bits of the result instead of the last 64 bits?

(12marks)

https://zhuanlan.zhihu.com/p/41716899

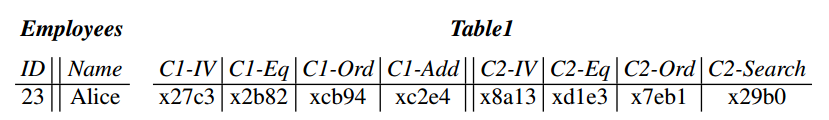
3. What is a purpose of using multiple layers of encryption in CryptDB approach to querying encrypted databases? Give an example of the situation where using three layers of encryption is justified. If needed you may either define some queries in SQL or describe them in the natural language.

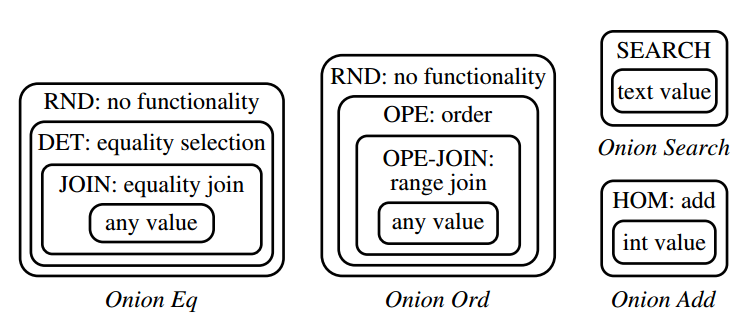
(20marks)

问题一、多层次加密方法查询加密数据库的目的是什么？

这主要是为了保证数据的安全性, 同时保证能够支持加密操作,是一种折中的设计.

问题二、举例说明三层加密是合理的





一开始所有的洋葱都是处于RND层次, 也就是加密等级最高的层次. 在这个层次, 没有DET和OPE的性质, 不能支持相应的操作. 如果某一次用户需要OPE或者DET的行为, 就对这一列数据进行解密, 解密到DET或者OPE的层次, 然后再进行处理. 这个剥洋葱的过程, 可以由MySQL-SERVER来完成, 因为这种解密不会暴露明文给服务器.

**参考文档**：<https://yiwenshao.github.io/>中的《Cryptdb原理概述(1)》

4. The company X has proposed a very fast and reliable biometric unlock system for their smartphones. It recognizes the owner’s face with different face expressions, open and closed eyes, etc with very high probability. What are the possible issues of using this system as the only authentication method granting acces to the smartphone?

(8marks)

生物信息具有唯一性和不可更改性，一旦泄露，则处于不可逆转状态。即人脸信息被窃取后不可能再恢复到保密状态

参考文档：<http://security.asmag.com.cn/news/201910/101028.html>

https://www.biometricupdate.com/201909/biometric-authentication-is-not-solving-the-password-problem

5. What could be the purpose of the following protocol, where both parties use RSA public-key algorithm, sk A is the secret key of A, pk B is the public key of B, and s is the secret message by B?

• Message 1. A → B : {{k} sk A } pk B ;

• Message 2. B → A : {s} k

Explain the rationale behind and the possible issue with this protocol. How can the issue be fixed?

(15marks)

6. The following schema for password based encryption has been proposed recently.

The password is split into two parts: the first part is short and easy to remember, the second part is randomly generated and kept secret - nobody knows it. When it comes to decryption the user enters his/her first part of the password, and the second secret part is brute-forced by the decryption algorithm and if successful, the ciphertext is decrypted.Discuss possible advantages and disadvantages of such a schema and compare it with possible alternative solutions.

(25marks)

peppering

参考资料：

<https://www.jianshu.com/p/55a29b701fa7>（这个写的不错）

# 对称加密技术 - PBE系列

PBE加密跳出了DES和AES的加密模式，综合对称加密、信息摘要算法的优势，形成了一个对称加密的特例。其实PBE加密并不是独创了一个新的算法，而是先使用MD（SHA）系列算法将口令和salt计算出一个key出来，然后用这个key作为DES（AES）的加解密key，这样DES和AES的key不便于记忆且需要计算机计算的问题就解决了。

## 算法特点

PBE算法，全称password based encrytpion，基于口令的加密，该算法不是新的算法，是融合算法，主要步骤如下：

* 首先用户输入口令，然后根据salt，采用信息摘要算法（MD5、SHA-256等）计算一个固定长度的摘要出来；
* 将该固定长度的摘要转换为对称加密算法（DES、AES）要求长度的密钥key；
* 使用key并采用对称加密算法对数据进行加密。

PBE没有密钥的概念，PBE使用口令替代了密钥。PBE算法是信息摘要算法和对称加密算法的融合，比如PBEWithMD5AndDES是MD5和DES的结合，类似的也可以反推。

### 口令和密钥的区别

* 口令：一般与用户名对应，是某个用户自己编织的便于记忆的一串单词、数字、汉字字符，口令的特点容易被记忆，也容易泄露和被盗取，容易被社会工程学、暴力破解、撞库等方式获取。
* 密钥：是经过加密算法计算出来的，密钥一般不容易记忆，不容易被破解，而且很多时候密钥是作为算法的参数出现的，算法对于密钥长度也是有要求的，因为加密算法的作用就是**利用密钥来扰乱明文顺序**；
* NOTE：口令不能代替密钥，但是可以用信息摘要算法将口令转换为密钥；

<https://stackoverflow.com/questions/16129562/pbes-advantage-over-des-3des-and-aes>（这个是说了些优缺点，用这个“password based encryption”做关键词搜索）

<http://www.crypto-it.net/eng/theory/pbe.html>（这个是PBE介绍如下）

## Password-Based Encryption (PBE)

Password-based encryption is a popular method of creating strong cryptographic keys.

The strength of the cipher depends on the strength of the secret key. A strong secret key must contain characters that are not easily predictable, thus the secret key cannot be simply derived from the user's password (because passwords are usually memorable subsets of ASCII or UTF-8 characters).

Password-based encryption allows to create strong secret keys based on passwords provided by the users. The produced key bytes are supposed to be as random and unpredictable as possible.

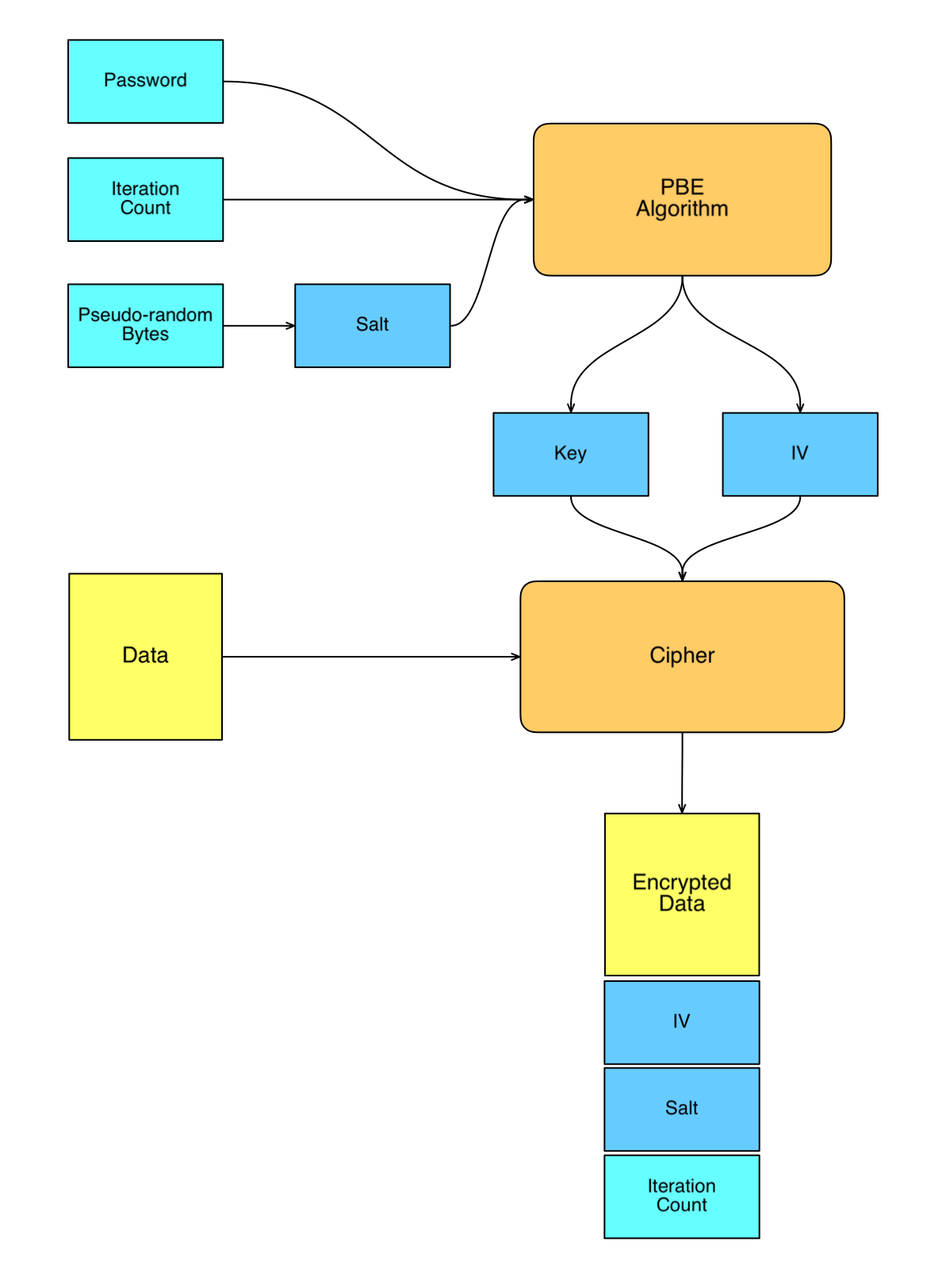
PBE algorithms use a user's password together with some additional input parameters:

* [salt](http://www.crypto-it.net/eng/theory/pbe.html#part_salt)
* [iteration count](http://www.crypto-it.net/eng/theory/pbe.html#part_iterations)

There are two popular PBE standards that describe how to convert password bytes into the secret key: PKCS #5 (supports ASCII characters) and PKCS #12 (which supports 16-bit characters).

In essence, they use a mixing function based around a [secure hash](http://www.crypto-it.net/eng/theory/one-way-function.html#hash) function which is applied a number of times (specified by an iteration count). After the mixing, the output bytes are used to create the key for the cipher (together with the initialization vector if needed).

A diagram of PBE algorithms



### Salt

The salt is a random number. It is supposed to prevent dictionary attacks. Without the salt, an intruder could use the same PBE algorithms and create a lot of keys for some popular phrases, often used as passwords. Adding a random value makes the combined input to the PBE algorithm completely random. It is no longer possible for the attacker to check all the likely PBE algorithm inputs.

Due to the fact that the salt is random, it is highly unlikely that the same salt would be reused twice, for multiple encryptions. The salt is not a secret value. It may be transmitted along with the ciphertext to the receiver.

Salt values are created by [pseudorandom number generators](http://www.crypto-it.net/eng/theory/pseudorandom-generator.html). Ideally, the length of the salt should be the same as the output size of the hash function that was used to create it.

### Iteration Count

The key derivation procedure may be made more complicated by running PBE algorithm many times. This would make the process of creating the secret key much more time consuming. Such a situation is certainly acceptable for the user, who has to perform the authentication procedure rarely and doesn't mind short delays. On the other hand, the attacker using [brute force attacks](http://www.crypto-it.net/eng/attacks/brute-force.html) and checking thousands of combinations would suffer significantly due to the increased time complexity.

Similarly to the salt, the iteration count may be transmitted to the receiver in the clear, along with the ciphertext.

It is recommended to use 1000 or more iterations to achieve a sufficiently good security level.