Contents

1. 课程介绍：

1.1. 概述（考试范围）. _

历史、本质特征、技术、每个技术干什么用的、应用（本质和特征比较重要：他就是个数据库、技术、协议、软件代码；去中心化（不依赖于中心化的情况下提供可信数据）、只增数据库；recoder 处理业务数据、转换成区块进行进一步处理）

1.2. 历史. _

比特币的出现：2008 年中本聪的电子支付系统（bitcoin）：去中心化（不依赖于任何的可信机构）（P2P 系统）（用户的匿名性）。并且提出了 blockchain 的概念。（只是货币）

2009 年 Bitcoin 的 network 开始运行。该系统开始运行并且到今天依旧在运行。

技术：2015 年以太坊出现：native cryptocurrency，可编程的，第二个或者第三个里程碑的事件，已经成为了一个重要的“区块链技术”，而不是普通的货币。

理论基础：然而金融相关的，缺少理论安全基础是非常危险的。15 年之前是没有理论基础，15 年对比特币的协议安全性进行了分析，意味着区块链的理论基础开始建立起来了。

1.3. 本质特征. _

一个区块链就是数据库（去中心化或者多中心化去大量数据维护的）、密码学技术为基础的对大量数据进行组织和维护的协议/数据库。

区块链技术是一种以密码学技术为基础，以去中心化或多中心化的方式对数据进行存储和管理，从而在**不依赖可信中心机构的情况下提供”可信数据“的技术**。

可信数据：不可伪造、不可篡改、不可否认、可溯源。

1.4. 应用场景. _

去中心化的智能合约

- 在那些依赖中心信任机构的应用中，去掉或降低中心机构所带来的成本、效率代价
- 银行清算等

提供可信数据：食品溯源、征信等

1.5. 区块链数据结构. _

- 处理数据：数据库的记录：records：业务数据、包括交易、事件等等
- 组织数据：区块：blocks：是有序记录的集合，每个区块只会写入数据库中各一次

- 串起数据：区块链：blockchain：只增数据库（增删改查），blockchain 只能有增和查，利用密码学哈希函数穿成一条链，实际上就是 block 的哈希链，每一个 block 包括了前一个 block 的哈希值。

数据被存储的方式是怎么做的：数据被存储为区块构成的链。

数据被添加的方式：什么数据能够被添加？（数据的添加是通过分布式共识、分布式数据库进行达成的，是为了去中心化），各个节点存储一个完整的数据库，并且通过各个节点之间运行分布式共识去决定什么数据能够被写入区块链。

所以数据的静态存储：哈希链；数据的动态添加：分布式共识。

1.6. 关键点. _

基本结构：分布式网络

本质：只增数据库（数据以 hash 链存储，通过分布式共识进行加入）

特征：在去中心化的情况下实现可信数据

区块链没有新技术，但是把现有的技术进行有机的粘合：包括 P2P 网络、分布式数据库、密码学、博弈论、分布式共识、智能合约等等。

区块链是一种分布式账本（Distributed Ledger）技术，是一种去中心化的数据库技术，是一种以密码学技术为基础的可信数据技术。

需要注意，区块链的分布式和正常的分布式数据库有所不同，正常的分布式数据库每个数据库节点互相信任，而区块链的分布式数据库是不互相信任的（只信任自己、既做服务器又做客户端），需要通过分布式共识来达成一致。

比特币没有用加密算法，比特币是一种数字货币，只用了 hash 等普通方法，加密应当意味着可以反推可破解。

1.7. 区块链的应用. _

公链：任何节点都可以参与操作（基本都是数字货币）

联盟链：企业政府等权威机构安全访问控制

私有链：只有某个组织可以参与系统的操作

1.8. 课程内容. _

区块链的超级简练简介、密码学简介、比特币介绍、共识协议的简单介绍、区块链的隐私保护（密码原语、环签名交易）、其他的一些区块链系统介绍

2. 技术概述：

2.1. 概述（考试范围）. _

交易、区块、链

2.2. 基本概念. _

区块链技术是一种以密码学技术为基础，以去中心化或多中心化的方式，对大量数据进行组织和维护，从而在不依赖中心结构的情况下提供可信数据。或者说是一个去中心化的或者多中心化的、公共的分布式总账，参与到系统上的节点可能不属于统一组织，彼此互不信任，区块链数据由所有节点共同维护，每个参与维护的节点都能获得一份完整记录的拷贝。

- 交易：对账本状态的改变，如添加一条记录
- 区块：记录一段时间内发生的交易和状态，是对当前账本状态的一次共识
- 链：由一个个区块按照发生顺序串联而成，可视为状态变化的日记记录

2.3. 基础框架. _

- 分布式数据库实现数据的分布式存储
- P2P 网络技术实现的动态组网
- 共识机制确保参与者数据的一致性
- 密码学中的数字签名和 Hash 算法等技术
- 经济博弈理论引导、规范理性参与者的行为
- （非常重要：应用推动技术的发展，前五点都是技术）可自动触发、自动执行的智能合约

2.4. 核心特性. _

- 去中心化/多中心化：

  分布式数据库是物理载体，区块链是逻辑载体，数据更新的去中心化是区块链去中心化/多中心化的本质。

- 可信数据：不可伪造、不可篡改、不可否认、可溯源。

  不可篡改（数据完整性）：数据库数据的不可篡改。

  不可否认（抗抵赖性）：交易数据的完整性验证。

- 智能合约

  区块链：实现智能合约的有效技术，带来了一场智能革命。

2.5. 基本属性以及分类. _

分布式、自治的、按照合约执行的、可追踪的

主流的分布式账本系统：基于 DAG、公证人、区块链技术的分布式账本

2.6. 挑战方向. _

安全机制、隐私保护、分布式共识协议、可扩展性及性能效率

3. 密码学基础：

3.1. 概述（考试范围）. _

引言：基础目标 Integrity/authenticity 实现这两个目标的目标原语：Hash、Symmetric、Asymmetric 等等... 数学计算：$227^{5791293}$, 作业题、群论或循环群、素数生成算法（多项式时间生成指数级大小素数）密码学假设：离散对数假设，CDH、DDH 假设（一定要知道）公钥加密：Syntax，有什么功能，保证什么事情、安全模型、大密钥空间元 CPA、CCA 的定义数字签名：和 hash 是最重要的，知道有什么算法、功能、安全保障 hash：函数，安全性，抗碰撞、第二原相抗碰撞（混合加密稍微了解即可）Public Key Infrastructure（PKI）：公钥密码学有一个公钥和身份绑定的问题，基本原理有 CA、数字签名、证书（掌握一下）比较 online key distribution server 与 CA 哪个更好?（公钥的 CA 确实比对称密码学的在线分发更是一种进步）

3.2. Cryptography Introduction. _

每年基本会考，包括完整性和机密性：

密码学的基本目标：安全通信：在不安全的信道上要能够实现机密性（攻击者不能获取消息内容）、完整性（消息内容和消息来源不被篡改）

实现机密性和完整性的是密码学的基本原语 (Primitives)，最基本的方案：

- 机密性：
  - **私钥/对称加密**

    收发双方使用相同的密钥，发送方加密信息后得到密文在不安全信道上传输，接收方使用相同的密钥解密得到明文。

    但是问题来了，对称密钥的问题是和不同的角色进行不同的密钥分发和建立，这样 N 个节点甚至有 $\frac{N(N-1)}{2}$ 的网络。

    因此比较好的方法：使用中心化的密码管理服务器，这样就可以通过中心化的密码管理服务器来分发密钥。session key：会话密钥 SeK，使用自己的密钥 K_1/K_2 和 SeK 进行加密后就可以获得对应的传输地址，然后就可以通过 session key 进行通讯。

    但是问题来了，中心化的服务（服务器坏了、失败了的鲁棒性）、系统里所有人的密钥都保存的问题（安全性泄漏）、服务器本身是否可信（可信性问题）

    因此提出了公钥密码学（是为了解决对称密码学的问题）
  - **公钥加密**

每个人都有公钥和私钥 p_k, s_k，任何人只要得到了对方的 p_k，就可以对消息 M 进行加密得到 $C = Enc(M, p_k)$ 后发送给对方，对方通过密钥 s_k 进行解密得到 $M = Dec(C, s_k)$。

公钥密码学解决了对称密钥分发的问题，但是引入了公钥和用户身份绑定的问题，因此需要 PKI（Public Key Infrastructure）来解决这个问题。

- 完整性：
  - **MAC（消息验证码）**

    收发双方共享密钥，发送方发送消息的时候使用 $(M, t = MAC_K(M))$，其中，t 是使用密钥 K 采取 MAC 算法对 M 进行处理得到的字段 t 后，将 M 与 t 拼接在一起得到的内容。接收方接收到消息后取出 (M, t) 后使用相同的密钥 K 计算 MAC 得到 t'，如果 $t = t'$ 则说明消息没有被篡改。

    要注意 MAC 算法并没有对明文进行处理！
  - **数字签名**

    每个人都有公私钥对，当你想要签名这个消息的时候，通过 s_k 对 M 进行签名得到 $S = Sign(M, s_k)$，对方通过 p_k 对 S 进行验证，如果验证成功则说明签名有效。

    好处：公开可验证、可转移的、不可否认性（非常之重要）。

    为什么 MAC 可以否认？它由于发送双方知道相同的密钥，使用相同的密钥进行 MAC 处理，无法判定是谁的。

    哈希函数 Hash function **（只是密码原语工具）**：哈希函数 $H : \{0,1\}^* \to \{0,1\}^n$

这几个密码原语只能实现具体一个特性，例如只能实现机密性或者只能实现完整性。数字签名只能用私钥去签名，而不是用私钥去加密，注意，加密是可逆的操作，签名是不可逆的。

3.3. 公钥密码学. _

- 定义：
  - 语法：函数和算法（功能性要求）
  - 安全：威胁模型和安全保证
- 用例：
  - 构建：具体的算法
  - 安全证明：某问题是困难的，那么它在这个构造方法下是安全模型

所以公钥密码学是在数学基础上进行开展的，是零碎的基础的数字基础。

3.3.1. Number Theory.

定义整除 divisors：$(b \neq 0), b|a \implies \exists m \in N, a = mb$，传递性、交换性、单位性

定义同余 congruence：$a \equiv b(\bmod m) \implies m|(a-b)$、自反性、交换性、传递性

定义模约运算 modular reduction：$a \bmod m = a - m\lfloor \frac{a}{m} \rfloor = r(a = qm + r, 0 \leq r < m)$，模约运算可以在式子的任何地方进行操作，可以拆分。

合数和素数 composite and prime：素数分解定理：任何一个大于 1 的整数都可以分解成素数的乘积，而且这个分解是唯一的：$a = p_1^{\alpha_1} p_2^{\alpha_2} ... p_t^{\alpha_t} (p_1 > p_2 > ... > p_t), 12250 = 7^2 * 5^3 * 2^1$，这个分解是唯一的。

最大公因数 GCD$gcd(a,b) = gcd(b, a\%b)$

模逆 modular inverse：$AB \bmod n = 1 \implies A^{-1} \bmod n = B$，需要注意，要存在模逆的前提是 B 和 n 互素。

找逆元可以考虑采取欧拉分解法，其中每次都进行对应的代换，其中每个代换都是前面一项的余数，代回来就可以求解逆元，可参照（3 区块链密码学基础.pdf）中国呢的求 911mod999 的逆元的例子。

欧拉 ϕ 函数：$\phi(n) = |\{a \in Z_n^* | gcd(a,n) = 1\}|$，其中 Z_n^* 是模 n 的乘法群，$\phi(n)$ 是 n 的互素的个数。并且该函数是可以累积乘的 $gcd(m,n) = 1 \implies \phi(mn) = \phi(m)\phi(n)$，其次对于质数 p 而言：$\phi(e^p) = p^{e-1}(p-1)$。最后 $\phi(n) = n(1-1/p_1)(1-1/p_2)...(1-1/p_k)$

费马小定理：质数 p，任何一个不被 p 整除的 a 均满足 $a^{p-1} \equiv 1(\bmod p)$ 因此如果 a, n 互素，那么 $a^k \bmod n = a^{k \bmod \phi(n)} \bmod n$

平方乘算法：计算 $11^{15} = 11^1 * 11^2 * 11^4 * 11^8$

3.4. Group Theory.

群论、群的定义、循环群的定义

3.5. 生成素数.

关键概念

- 安全参数：算法会有复杂性，所有的计算复杂度都用 n 来衡量，攻击者进行攻击时，成功的概率也是使用 n 来进行衡量的。
- 有效算法：算法的运行复杂度至多是 n 的多项式。
- 可忽略概率：可忽略函数 $f : Z^+ \to \{0\} \cup R^+$，对于任何多项式 $p(n)$，存在 n_0，当 $n > n_0$ 时，$f(n) < \frac{1}{p(n)}$，也就是说，$f(n)$ 是一个非常小的函数，可以忽略不计。

当我们说一件事情发生的概率是可忽略的，那么就意味着它是不可能发生的。

生成素数 Generate Random Primes：

输入：长度为 n，参数为 t

输出应当是一个 $n-bit$ 的素数：算法，在 0,1 数列中随机选择一个长度为 n-1 的 01 串，并且令 p' = p+'1'，如果 p' 是素数，返回 p，否则停止。

因此应当尝试多少次？首先，$n>1$ 的情况下 $n-bit$ 的素数至少有 $\frac{2^n-1-(2^{n-1})}{/}3n$ 个

那么随机挑素数不为素数的概率是 $(1-\frac{1}{3n})^t=((1-\frac{1}{3n})^{3n})^n\leq(e^{-1})^n=e^{-n}$ 实际上是一个可忽略的概率，因此是能够找到素数的。

所以提出了一个 Miller-Rabin primality test 去检验素数。

因此我们只需要：长度 n 的素数，我们从 1 到 $3n^2$ 每次都随机生成一个 01 串 (n-1) 位的数字再补 1，然后进行 Miler-Rabin test 就可以得到输出是否是素数的判断，如果是直接返回；如果 $3n^2$ 后仍旧没成功，返回 “失败”。因此通过一个有效算法（经过多项式时间）生成了一个 n-bit 的素数，即大素数。

3.6. 密码学的困难假设. _

群生成算法 GenGroup：基于 n 的输入安全参数，提供一个**循环群** G 的描述、它的阶数 $|q|=n$ 并且一个生成但愿 $g\in G$。描述提供了一个群的说明方式，群元素和二进制字符串之间的关系；并且要有有效的二元运算去计算群元素。则根据有效的群元素算法可以生成一个有效的随机采样算法（例如简单选择一个 $x\in Z_p, h=g^x \implies h\in G$ 即我们采样得到了 h），如果 x 选择足够大的情况下，那么就通过平方相乘法就可以很好地把数字计算出来，例如 $2^{15}=2^1*2^2*2^4*2^8$，因此可以通过平方相乘法进行计算。

3.6.1. 离散对数问题 *(DL).* _

开始定义循环群上的离散对数（Discrete Logarithm）：考虑到 $G=g^0,g^1,...,g^{q-1} \implies \forall h\in G, \exists! x\in Z_q, h=g^x$，其中 $x=\log_g h$ 就是离散对数。

离散对数假设：在循环群 G 上，有群生成算法 GenGroup、算法 A、参数 n，离散对数判定算法为：

运行群生成算法去获得 (G,q,g) 后，把 (G,q,g,h) 输入 A，A 输出了一个 $x\in Z_q$，如果 $g^x=h$ 则算法输出 1 否则输出 0。

所以离散对数问题为：给定 $(G,q,g)\leftarrow GenGroup(1^n)$ 以及一个标准的 $h\in G$，能够找到 $x\in Z_q s.t. h=g^x$

那么，一个在**群生成算法上的问题**是**困难的**，是对于离散对数问题的成功求解（多项式时间算法）概率是可忽略的。

3.6.2. Diffie-Hellman 假设 *(CDH).* _

同样针对的是群生成算法 GenGroup、算法 A、参数 n，则 Computational Diffie-Hellman(CDH) 实验定义如下：

运行群生成算法去获得 (G,q,g) 后，选择标准的 $x_1,x_2 \in Z_q$ 并且将 (G,q,g,g^{x_1},g^{x_2}) 输入 A，A 输出了一个 h，如果 $h=g^{x_1x_2}$ 则算法输出 1 否则输出 0。

从而得到 CDH 定义问题：给定 $(G,q,g) \leftarrow GenGroup(1^n)$ 以及标准的 $h_1,h_2 \in G$，能够找到 $h \in Gs.t.h=g^{x_1x_2}, x_1=\log_g h_1, x_2=\log_g h_2$

那么，一个在**群生成算法上的问题**是**困难的**，是对于 CDH 问题的所有多项式时间算法成功求解概率是可忽略的。

3.6.3. Decisional Diffie-Hellman(DDH) 问题. _

这是一个判定问题，DDH 的定义如下：

运行群生成算法获得 (G,q,g) 后选择两个标准的元素 $x_1,x_2 \in Z_q, h_1=g^{x_1}, h_2=g^{x_2}$；然后随机选择一个 $\{0,1\}$ 的数 b，如果 $b=0$ 随机选一个 $T \in G$，否则选择一个 $T=g^{x_1x_2}$；然后让算法 A 检测 (G,q,g,h_1,h_2,T)，A 输出 b'，如果 $b'=b$ 则 A 成功，A 成功的概率为 $Adv_{A,DDH,GenGroup}=|\Pr[b'=b]-\frac{1}{2}|$。

简单来说，DDH 问题即给定 $(G,q,g) \leftarrow GenGroup(1^n)$ 以及标准的 $h_1,h_2 \in G$ 以及一个元素 $T \in G$，判定 $T=g^{x_1x_2}, x_1=\log_g h_1, x_2=\log_g h_2$ 是否成立。

那么，一个在**群生成算法上的问题**是**困难的**，是对于 DDH 问题的所有多项式时间算法成功判断概率是可忽略的。

3.6.4. 群生成算法 *GenGroup.* _

群生成算法 GenGroup：基于 n 的输入安全参数以及长度 $l(n)$，提供一个**循环群** G 的描述、它的阶数 $|q|=n$ 并且一个生成单元 $g \in G$。描述提供了一个群的说明方式，首先生成一个 $n-bit$ 的 q，然后生成一个 $l-bit$ 的质数 p 使得 $q|(p-1)$，然后选择一个标准的 $h \in Z_p^*, h \neq 1$，设定 $g:=h^{\frac{p-1}{q}}$，则 (G,q,g) 就是一个循环群。

写一个群，如果是写的一个素数群例如 $q=11$ 的素数群，此时就需要令 $p=23$ 后去进行 $11|(23-1)$ 再去生成。（以及注意作业中对于这块内容的要求和涉及）

注意：在这个 GenGroup 群生成算法上的 DL、CDH、DDH 问题是困难的，在其他简单的生成群算法上（暂且不能保证是困难的）；当然还有其他群，例如椭圆曲线上的曲线生成算法 ECDSA 等等（安全性也比较好）

3.6.5. 公钥加密. _

- 语法 Syntax：存在多项式时间算法 $(KeyGen, Enc, Dec)$
  - $KeyGen \rightarrow (pk, sk)$

  - $Enc(pk, m) \to c$
  - $Dec(sk, c) \to m$

- 正确性 Correctness: $\forall m \in M, Dec(sk, Enc(pk, m)) = m$
- 安全模型：攻击者不能突破这个设定的场景。
  - Security Guarantee: 安全保证定义了这种情形防治了什么样的攻击。

    安全保证应当确保攻击者恢复密钥是不可能的（这是必要条件，如果是充分显然不对，因为 $Enc(k, m) = m$ 这种方案虽然无法让攻击者恢复，但是并没有保护到信息本身）；

    安全保证应当保证攻击者没法从密文中生成明文（这还是必要条件不是充分条件，因为例如入侵公司数据库能够获取到 10% 员工的薪水虽然有 90% 无法获得，但是这种泄漏体量也是我们无法接受的）；

    安全保证应当保证攻击者不能从密文中获取任意一个字符的明文信息（这还是不够细，例如一个薪资数据库能够判断薪资是否高于 100000 但是没有额外信息，这种泄漏也是我们不允许的）；

    所以，**一个密文应当不泄漏明文的任何额外信息，这就是安全保证的定义**。
  - Threat Model: 威胁模型描述了对手的力量，即攻击者被假定能够实施什么行动。

    有四种攻击：唯密文攻击（只能窃听到密文）、已知明文攻击（能看到明文进行攻击）、CPA（选择明文攻击：攻击者给定明文，被攻击者给定密文）、CCA（选择密文攻击：攻击者攻击给定密文能够获得选择明文）

- Kerckhoff's Principle

  方案的安全性只能依赖于密钥的安全性，设计一定是公开的而不是保密的。因为方案本身的保密实行起来是非常不方便的，并且如果泄漏了则代价是非常大的，保密算法本身是非常不合适的。因此使用密钥的保密性一定比方案/算法本身的保密性来的好。只有精灵过公开分析的密码算法，才可能是“好的（安全的）”算法。

- Sufficient Key-space Principle

  一个密码算法方案要抵抗 Brute-force 的暴力破解，任何一个安全加密方案要拥有一个足够大的密码空间使得暴力破解是不可行的。（注意，这是一个必要条件）

- CPA 安全：选择明文攻击

  考虑公钥加密的一个实验：提供 $(KeyGen, Enc, Dec)$，对手 A 和安全参数 λ，通过 $KeyGen(\lambda) = (p_k, s_k)$，并且对手 A 已知 λ, p_k，此时竞猜过程中对手 A 采取操作：A 提供

两个明文信息 $m_0^* \neq m_1^* \in M_s$，随机选择 $b \in [0,1]$，然后得到了密文 $c^* = Enc(p_k, m_b^*)$，然后 A 输出 b'，如果 $b' = b$ 则 A 成功，A 的优势为 $Adv_{A,CPA,Enc}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|$。

CPA 安全的定义：对于任何多项式算法的对手，我们进行这个实验操作得到的优势应当都是可忽略的，即密文没有给明文带来额外的信息。

- CCA 安全：选择密文攻击

  考虑公钥加密的一个实验：提供 $(KeyGen, Enc, Dec)$，对手 A 和安全参数 λ，通过 $KeyGen(\lambda) = (p_k, s_k)$，并且对手 A 已知 λ, p_k。在探索阶段，对手 A 具有解密的功能（获得了解密机的能力），即 A 可以对密文 c 进行解密得到明文 $m \to Dec(sk, c)$。在挑战阶段：A 提供两个明文信息 $m_0^* \neq m_1^* \in M_s$，随机选择 $b \in [0,1]$，然后得到密文 $c^* \to Enc(k, m_b^*)$（挑战密文）；在第二个探索阶段，A 再次进行对密文的解密，但是此时选择的 $c \neq c^*$；在第二个挑战阶段，A 输出一个 b' 作为猜测；如果 $b' = b$ 则 A 成功，A 的优势为 $Adv_{A,CCA,\Pi}(\lambda) = |\Pr[b' = b] - \frac{1}{2}|$。

  CCA 安全的定义：对于任何多项式算法的对手，我们进行这个实验操作得到的优势应当都是可忽略的，即密文没有给明文带来额外的信息。

  核心区别：CCA 比 CPA 强，是一种攻击者的访问。CPA 只能访问加密机，CCA 可以访问解密机。

- 一个具体的 PKE（公钥加密）场景：EI Gammal Encryption

  令 GenGroup 定义一个多项式时间内的算法并且能够基于输入的 1^n 输出一个循环群，它的阶数 q 以及一个生成元 $g \in G$

  - $KeyGen(1^\lambda) \to (pk, sk), pk := (G, q, g, h), sk = (x)$
  - $Enc(pk, m) \to c, (c_1, c_2) := (g^y, m \cdot h^y)$
  - $Dec(sk, c) \to m, sk = (x), c = (c_1, c_2) \in G^2, m := c_2 * (c_1^x)^{-1}$

  定理：如果 DDH 问题是困难的，那么 EI Gammal Encryption 是 CPA 安全的。（其中要注意，DDH 是最简单的，CDH 其次，DL 是最难的）（实际应用中会使用足够大的安全参数，例如 $\lambda = 128$ 才能足够安全）

  EI Gammal Encryption 不是 CCA 安全的，需要注意。（作业问题）

3.7. 数字签名. _

使用密钥对信息进行签名，对方通过公钥进行验证。

- Syntax:

  - $KeyGen \to (pk, sk)$
  - $Sign(sk, m) \to \sigma$ 这就是签名，而不是什么密文
  - $Verify(pk, m, \sigma) \to \{0,1\}$，其中 σ 是签名

- Correctness: $Verify(pk, m, Sign(sk, m)) = 1$
- 安全性：攻击者是没有能力去伪造（消息、签名）对

  安全模型：攻击者只要能够输出一个与消息不同的消息并且有对应的签名，则说明成功了。

  所以对任何多项式时间算法都没有一个伪造（消息，签名）对的数字签名方案。

- 威胁模型：攻击者看到公钥、还能看到签名原机
- 具体的签名方案：DSA/ECDSA
  - $KeyGen(1^\lambda) \to (pk, sk), pk := (G, q, g, y = g^x), sk := (x) \in Z_q$，其中要注意的是有两个函数可以生成 Z_q，例如 $H : \{0,1\}^* \to Z_q, F : G \to Z_q$
  - $Sign(sk, m) \to \sigma, \sigma := (r, s), k \in Z_q, r = F(g^k), s := k^{-1}(H(m) + xr) \bmod q$
  - $Verify(pk, m, \sigma) \to b \in \{0,1\}, b = 1 \Leftrightarrow r = F(g^{H(m)s^{-1}} y^{rs^{-1}})$

  安全性：DSA 和 ECDSA 都是安全的（当 H 和 F 都是正确的建模方式），例如

  - DSA：G 是 Z_p^* 的子群，$F(I) := I \bmod q$
  - ECDSA：G 是椭圆曲线上的点，$F((x, y)) := x \bmod q$

3.8. 哈希函数. _

多项式时间，把一个任意长的 01 串映射到固定长度的 01 串

3.8.1. 安全模型. _

一个哈希函数是抗碰撞的：即没有多项式时间的攻击者能够找到 $x \neq x' \in \{0,1\}^* \implies H(x) = H(x')$

第二原象安全（目标抗碰撞）：提供一个 x，多项式时间找不到满足 $x' \neq x, H(x) = H(x')$

目标原象抵抗：提供一个 y，多项式时间找不到满足 $H(x) = y$ 的 x

抗碰撞可以推导出第二原象安全，第二原象安全可以推导出目标原象抵抗。

3.9. Hybrid Encryption 混合加密. _

公钥加密解决了对称加密之间的密钥分发问题，但是运行速度变慢了，因此有混合加密的想法，进行混合。利用公钥加密解决密钥分发问题，对称加密解决数据传输效率问题。

核心方案：随机选择对称加密的密钥 K，用接收方（Bob）的公钥进行加密后，再用对称密钥 K 加密，然后发送给 Bob，Bob 先用私钥恢复出来 K，再用对称加密的方法回复出来数据即可。

KEM: 数据封装通过输入 pk 得到 (c, K)，其中 c 是密文，K 是对称加密的密钥；数据解封通过输入 sk 和密文 c 得到 m。

3.10. PKI: Public Key Infrastructure. _

使用者如何确保公钥、签名是对方的？因此采取了数字证书的概念（Digital Certificate）

3.10.1. 数字证书 *Digital Certificate.* _

假设 CA 是被设置为可信的，CA 可以对公钥进行签名，然后把公钥和签名一起发送给用户，用户可以通过 CA 的公钥进行验证，如果验证成功则说明是可信的。核心概念：CA 对公钥和身份的绑定进行了背书。

$Cert_A = (ID_A, PK_A, expiry - date, Sign_{CA}(ID_A, PK_A, expiry - date))$

注意：

- 浏览器如果写证书，是需要认证这个证书，意味着如果你被写入了，那么就被攻克了。
- 一个证书不会比浏览器的安全更高（浏览器如果被黑了那证书也同样危险了）。

证书是具有有效期的，Validity Period（保证 CA 是不做坏事的可信性，要让它赚钱），因此需要定期更新证书。

同样，CA 是一个可信的第三方。

3.10.2. 公钥基础设施 *PKI.* _

需要在一套基础设施上进行新人，例如寡头模式：可以信任多个 CA（以避免垄断独裁的情况）。PKI 我们还是需要相信一个 CA，CA 做坏事的概率比较小，所以技术缺点是比较小的。

在线密钥分发服务和 CA 相比：前者可以看到所有的通信、并且由于他们有我们的密钥，是可以做这种坏事的：而 CA，我们只需要确认它不会给钓鱼网站发 CA，CA 一旦做了坏事，是可以验证的，留下了不可否认的证据。

3.11. Diffie-Hellman Key Exchange Protocol.

利用了离散对数问题，如何两个人交换信息? 例如 Alice 传递 $g^a \bmod p$，Bob 传递 $g^b \bmod p$，则二人共享了 $K = g^{ab} \bmod p$，这个过程是安全的。

然而如果中间主动攻击人拦截了的话就是会变得不安全了。g^{at},g^{bt}，中间人转发了 $g^t \bmod p$

4. 比特币（1）

4.1. 概述（考试范围）. _

介绍比特币是为了介绍区块链的技术精髓比特币的账本：trasaction base, UTXO model, Digital signature 的使用、隐私保护问题（隐私保护的机制：用公钥或者公钥的 hash 作为 owner，但是这样会导致一个问题：如果一个人有多个账户，那么这个人的所有账户都会被关联起来，所以比特币的隐私保护是一个很大的问题）比特币的交易结构：transaction 的 metadata、input(s)、output(s)、交易的有效性（三个基本规则，不能超花、不能双花、需要签名）以及检查规则

4.2. 账本. _

比特币：支付系统的分布式账本

4.2.1. account-based ledger. _

以太坊通过这种基于用户的账本来表示账本里的具体交易形式（账本里的交易包括了所有交易记录，每一次交易记录都会更新账户的盈余状况，这种账本的形式是账户的形式，因此称为 account-based ledger）

4.2.2. transaction-based ledger. _

比特币使用的记账方法，不需要额外维护反应状态的表，系统的 records 就是交易本身，核心的业务对象就是 Transaction Output(TXO)，交易对象是 1 枚硬币，本身用交易描述了每一次的关系，每个交易要去指向前面的某个 TXO 来产生新的 TXO。这个时候，由于每次采取的都是前向的 TXO，所以不需要维护 account-based 的表，只需要检查前向的 TXO 是否满足即可成立。注意，每个 TXO 只能被消耗一次（10 两银子），因此每笔交易只能消耗未被消耗掉的 TXO（Unspent TXOs），这种就叫做 UTXO 模型。

4.2.3. Bitcoin. _

比特币是基于交易的分布式账本，每一个记录都是一笔交易，每一个交易输出（TXO）都表示了一枚 coin，包括了 (owner, value) 的信息并且每笔交易都消耗现存的 TXOs/coins 去生成新的 TXOs/coins。注意，每个 TXO 只能被消耗一次（10 两银子），因此每笔交易只能消耗未被消耗掉的 TXO（Unspent TXOs），这种就叫做 UTXO 模型。

注意到，这是币值的流动链（每笔交易可以有对应的 TX 流动链）而不是区块链，要注意区块链是技术，比特币是一种方式。

比特币中，公钥是作为一个 owner 的证明。如何证明你是 public key 的 owner：生成 public key 的数字签名。用签名对消费 TXO 进行验证 $Verify(pk, tx, \sigma) = 1$。在区块链中，

密钥是唯一证明能够花费 TXO 的方式。

Bitcoin 有两种交易：一种是 Coinbase Transaction，凭空生成新的 coin，另一种是普通交易，消耗已有的 TXO 产生新的 coins。

隐私实际上是非常差的（因为这些数据都是可追踪的，签名都是可追踪的，实际上只是化名，化名的隐私非常低的）

接下来：hash 值上的币址：钱放在 hash 值上，然后根据 hash 值进行花费。

hash 做地址，能提供更多的可能性，例如智能合约也是有可能的。

5. 比特币（2）

5.1. 概述（考试范围）. _

开始组织成区块：Merkle Tree（对数级复杂度组成的树，使用 Hash pointer 建立出来的）Hash Pointer，拿数据的 hash 值作为唯一索引，block chain 也用 hash pointer 串起来的 hash chain，交易指向前面某一个 TXO 也是使用了 hash pointer；区块通过 Merkle Tree 建立起来，是整个数据的链；讲 bitcoin 挖矿的基本原理、怎么去挖矿，知道它如何利用 hash 函数的密码学实现工作量大证明，通过 hash 函数寻找小范围 hash 值的原像；（很重要：How a transaction is recorded on blockchain 在课程视频里有）；比特币的共识（先到原则、最长链原则等等）；

5.2. 区块. _

5.3. 区块链. _

6. 比特币（3）

6.1. 概述（考试范围）. _

矿池（mining pool）、为了讲基于 PoW 的机制、Block Discarding Attacks（区块丢弃攻击：降低系统有效算力，无效算力还能去窃取别人的算力等等）、矿池优劣性、分叉攻击（通过分叉实现的双花攻击、一定要掌握）、自私挖矿（selfish mining 利用比特币 PoW 共识实现的攻击）；钱包（整个钱包都是很重要的）什么叫钱包、确定性钱包的基本算法、好处、基本原理、潜在的漏洞、分层确定性钱包（summary 很重要）

6.2. 比特币矿池. _

6.3. 比特币的一些攻击. _

6.4. 比特币脚本的应用. _

6.5. 比特币密钥管理：钱包. _

6.6. 简单对比特币和区块链的总结. _

7. 共识协议：

7.1. 概述（考试范围）. _

内容比较单纯，简单介绍共识的几个 COP、IOP 定理，PoW 协议，CAP 定理、衡量共识协议的 CP 定理共识协议、比特币的重要共识协议（至少知道一些特点，PoW 的优缺点、PBFT 的 summary 也需要知道、细节不需要知道）

8. 隐私部分（1）

8.1. 概述（考试范围）. _

比特币的隐私问题很严重（为什么很严重）、coin-mix（混合去隐私，怎么工作的），coin-shuffle（考虑 coin-join 和 coin-shuffle 的区别，去中心化不需要第三方服务的）是比较容易理解且有趣的且二者的优缺点以及对比

9. 隐私部分（2）

9.1. 概述（考试范围）. _

密码原语（隐形地址：第一个例子有毛病为什么有毛病、第二个例子怎么改进、改版、低三个例子是第二个改版，已经比较完美了，如何到达完美的，当然还是有一定的毛病的）环签名（Syntax)、可连接环签名的特点，能够保证什么事情、保证什么安全？承诺 Commitment and range proof（非常重要)：承诺、承诺对应的三个算法、两个安全性质、需要给一个 range proof，如何生成的

10. 隐私部分（3）

10.1. 概述（考试范围）. _

Monero 币用的技术，三要素保护起来了；Commitment and range proof 非常重要，怎么生成，怎么用一个很简单的方法（区块链用密码学实现矿工不获得任何信息的情况下获取验证内容）这块非常重要、讲了 Monero 币的生成过程（不重要）、但是密码原语是非常重要的(Commitment and range proof)

11. 隐私部分（4）

11.1. 概述（考试范围）. _

更深层次的未来可以做什么，提到了隐形地址的三步走（仍然有毛病：给你两枚 coin，把其中一枚 coin 的密码偷了可以花掉另一枚的等等），比较虚

12. 考试细节：

12.1. 考试：．_

全是大题，没有选择判断题，类似作业题，都是论述题

12.2. 去年题目：．_

1. 抗碰撞性、原相安全，抗碰撞证明第二原相安全
2. 账本是什么，数据怎么组织，系统怎么运行
3. C 一致性 A 可达性 P 不连通性容忍不可能三角是什么意思
4. 区块链共识协议的关键因素（能容纳的节点和支持的网络规模）
5. 拜占庭问题和特征
6. 洗钱怎么洗（洗币服务器怎么隐藏付款者和金额）
7. 隐形地址算法，怎么把发送、接受者藏掉
8. 数学题（可能是同余）
9. 你们小组讲的哪篇文章请你归纳一下

12.3. 今年题目：．_

. 数学题（还是大整数大素数同余，类似作业）

. Hash 指针以及怎么用在区块链网络中的

. Merkle Tree 以及如何识别链路

. 怎么达成一笔交易、怎么共识一个区块

. 一个新节点从零开始接入区块链网络的过程

. 抗矿池算法、分叉攻击、自私挖矿三个条目选两个

. Commitment and RangeProof 怎么分析，给出具体验证方法

. 你们小组讲的哪篇文章归纳一下、以及随机某四个小组的文章选一篇请你归纳一下