作业三

(密码学与网络安全课程报告)

姓　　名：肖文韬

学　　号：2020214245

专 业 方 向：电子信息（计算机技术）

邮　　箱：xwt20@mails.tsinghua.edu.cn

二〇二一年三月二十六日

第 1 章　作业内容

1. 在 RSA 中，如果只知道公钥 e 和模数 N，不知道 p、q，求的私钥 d 的计算复杂度是多少或多大数量集？如果已知 p、q 呢？请给出证明。

   解：

   如果 n 可以被质因数分解，d 就可以算出，也就意味着私钥被破解。可是，大整数的因数分解，是一件非常困难的事情。暴力枚举的话就是 $O(N)$，不过因为 N 特别大，例如目前常用的 2048 bits 的 RSA，N 能达到 $2^{2048} \approx 10^{616}$ 的规模。根据维基百科 Prime number theorem 条目，质数在自然数中的分布大概是 $\pi(M) \sim \frac{N}{\log_e(N)}$，所以暴力枚举质数的复杂度大概为 $O(\frac{N}{\log_e(N)}$。

   如果已知 p、q，我们需要使用扩展欧几里得算法来计算 $de \equiv 1(\bmod \phi(N))$（$e, \phi(N) = (p-1)(q-1)$ 均已知）中的 d，所以复杂度为 $O(\log e \log \phi(N))$.

2. 在 RSA 中：1) 给出 n=3937 和 e=17，求私钥 d。2) 给出 n=253 和 e=3，求私钥 d，并对明文 m=165 进行加密求出密文 c。

   解：

   1) 由 $de \equiv 1(\bmod \phi(n))$，等价于 $de + k\phi(n) = 1 = \gcd(d, e)$ (关于 d 和 k 的二元一次方程)。其中 $\phi(n) = (p-1)(q-1), n = pq$（$pq$ 为两个质数），可以通过暴力枚举得到 $3937 = 31 \times 127$，故 $\phi(n) = 30 \times 126 = 3780$。该式可以用扩展欧几里得算法求解，得到 $d = 667, k = -3$。

   2) 同理，使用扩展欧几里得算法求解 $de + k\phi(n) = 1$（其中 $\phi(n) = \phi(253) = \phi(11 \times 23) = 220$），得到 $d = 73$。对于加密过程，即求 $c = m^e \bmod n$，又因为 $(a \times b) \bmod p = (a \bmod p) \times (b \bmod p)$。故 $c = 165^3 \bmod 253 = 110$。

3. 设 E 是由 $y^2 \equiv x^3 + x + 6 \bmod 11$ 所确定的有限域 Z_{11} 上的椭圆曲线，设基点 $P = (2, 7)$, 保密的私钥 $d = 7$。1）计算 $2P = P + P$ 2) 若公钥 $Q = 7P = (7, 2)$, 假设明文 $m = (5, 6)$, 计算对应的密文。

   解：

   因为课上讲的比较少，可以不做。

4. 简述杂凑函数有哪些功能? 应满足哪些条件?

   解：

   杂凑函数即哈希函数（hash），是一种将任意数据压缩成摘要（哈希值）的单向函数。哈希值通常用一个短的随机字母和数字组成的字符串来代表。

   一般来说需要满足下列条件：

(a) Hash 可用于任意大小的数据块。

(b) 输出为固定长度的消息摘要。

(c) 单向性。给定一个输入 M，一定有一个 h 与其对应，满足 H(M)=h，反之，则不行，算法操作是不可逆的。

(d) 抗碰撞性。给定一个 M, 要找到一个 M’满足是不可 H(M)=H(M’) 是不可能的。即不能同时找到两个不同的输入使其输出结果完全一致。

(e) 低复杂性：算法具有运算的低复杂性。

5. 什么是陷门单向函数? 陷门单向函数有何特点? 如何将其应用于公钥密码体制中?

解：

一个函数是单向陷门函数，是指该函数正向计算是易于计算的，但求它的逆是不可行的，除非再已知某些附加信息。当附加信息给定后，求逆可在多项式时间完成（NP 问题）。

研究公钥密码算法就是要找出合适的陷门单向函数。例如背包密码体制, RSA 算法都是利用了逆运算在不附加信息时计算不可行（NP 问题）的基础上实现 PKI 的。公钥即是陷门单向函数的正向计算过程，计算比较简单，而且可以公开。而在不知道私钥的情况下逆向计算就是单向陷门函数的逆向计算，其计算不可达性保证了加密算法的安全性。而私钥就是附加信息，在知道私钥的情况下，单向陷门函数的逆向计算也变成了很容易计算的 P 问题。

6. 思考题: 如何衡量一个密码系统的安全性?Diffie-Hellman 算法的安全性如何?

解：

一个密码系统的安全性主要与这些方面有关:

(a) **无条件安全性**。即使密码分析者拥有无限的计算资源和密文，都没有足够的信息恢复出明文，那么这个算法就具有无条件安全性。香农曾经证明了一次一密乱码本（one-timepad）是不可破解的，此种情形下，密钥流是完全随机的、与明文相同长度的比特串，即使给出无限多的资源仍然不可破。虽然其具有理论上的绝对安全性，但考虑到密钥传输的代价，它又是不实用的。

(b) **计算安全性**。在实际中，无条件安全的系统是不存在的，我们通常所说的算法安全性，就是指算法的计算安全性。如果算法用现在或者将来的可用资源都不能破译，那么，这个算法被认为是计算安全的。在实际应用系统中，当破译某个密码算法的所需的计算时间或成本费用远远超过信息有用的生命周期或者信息本身的价值时，那算法破译本身就没有意

义了。这时也可以认为该算法具有计算安全性。

(c) **可证明安全性**。算法的安全性可规约为某个经过深入研究的数学难题（如大整数素因子分解、计算离散对数等），数学难题被证明求解困难。不过，当量子计算出现之后，针对目前使用的 RSA、DH 和 ECC 等公钥算法的计算安全性不再有理论保证，因此，密码界正在开展抗量子密码研究，以期在量子计算机成为现实攻击工具之前找到新的出路。

Diffie-Hellman 算法的安全性:

偷听者（”Eve”）可能必须通过求解迪菲－赫尔曼问题来得到 g^{ab}，该问题又称为离散对数问题，其求解是非常困难的。

1）但是如果 Alice 和 Bob 使用的随机数生成器不能做到完全随机并且从某种程度上讲是可预测的，那么 Eve 的工作将简单的多。还有一个 DH 早期版本的安全性问题是中间人攻击，攻击者 Eve 可以在 Alice 和 Bob 通讯的过程中，截获 Alice 和 Bob 的消息并伪造假消息，使得 Alice 计算出来的密钥实际上为 Alice 和 Eve 之间协商的密钥，Bob 计算出来的密钥实际上是 Blob 和 Eve 之间协商的密钥。密钥交换不能抵御上述攻击，是因为没有对通信的参与方进行认证。所以 TLS 就引入了认证机制来进行防御。

2）另外一个安全性问题就是并非所有 Diffie-Hellman 参数都可以“安全”使用。Diffie-Hellman 的安全性取决于称为离散对数问题的特定数学问题的难度。如果可以解决一组参数的离散对数问题，则可以提取私钥并破坏协议的安全性。一般来说，使用的数字越大，解决离散对数问题就越困难。因此，如果选择较小的 DH 参数，则会遇到麻烦。2015 年的 LogJam 和 WeakDH 攻击表明，许多 TLS 服务器可能被欺骗使用 Diffie-Hellman 的小数字，允许攻击者破坏协议的安全性并解密对话。Diffie-Hellman 还要求参数具有某些其他数学属性。2016 年，Antonio Sanso 在 OpenSSL 中发现了一个问题，其中选择的参数缺乏正确的数学属性，导致另一个漏洞。TLS 1.3 采用固定路由，将 Diffie-Hellman 参数限制为已知安全的参数。但是，它仍然有几个选择; 只允许一个选项使得在以后发现这些参数不安全的情况下更新 TLS 非常困难。