写给程序员的TPM入门指南

第一章

关于RSA加解密算法

Copyright: 青岛中怡智能安全研究院

程序员使用RSA函数库时, 一般不会直接触及底层的数学概念. 对于从未接触过RSA的开发人员, 我们也不强制要求每个人必须深入理解e、d是如何生成的. 但是作为开发人员, 理解这些基础概念仍然是有益的. 就好比在完全不理解汽车发动机/变速箱/刹车片中的机械结构的情况下, 新手开车上路虽然也可以开得很6. 但是只有当老司机们深入理解了汽车的内部原理后, 半道上车子出了毛病, 才有底气(思路)敢于去推导故障的原因.

问题一

RSA算法是个什么鬼？

问题二

RSA算法中用到的公钥和私钥, 葫芦里面装的什么鬼？

简单的回答是公钥里面装的是两个数字(n,e), 私钥里面装的是两个数字(n,d)

问题三

RSA公钥算法中有一堆纯数学概念, 能否一一列举？

公钥指数e、私钥指数d、模数n

n=p\*q、欧拉公式φ(n)=(p-1)\*(q-1), 其中因子p和因子q

1. 随机选择两个不相等的质数p和q（例如爱丽丝(Alice)选择了61和53）

2. 计算p和q的乘积n=p×q=61×53=3233

3. 根据本文“欧拉函数”介绍过的公式  
φ(n)=(p-1)(q-1)  
将p和q代入计算n的欧拉函数值  
φ(n) =(p-1)(q-1)=(61-1)×(53-1)=60×52=3120

4. 随机选择一个整数e，条件是1<e<φ(n)，且e与φ(n)互质  
爱丽丝可以在1到3120之间，随机选择一个17

对于二进制的电子计算机系统, 公钥指数e取下列质数3, 5, 17, 257, 65537, 4294967297时,进行二进制运算最方便:

1 = (01)2进制

3 = (11)2进制 = 1\*1+1

5 = (101)2进制 = 2\*2+1

17 = (1001)2进制 = 4\*4+1

257 = (1 0000 0001)2进制 = 0x0101 = 16\*16+1

65537 = (1 0000 0000 0000 0001)2进制 = 0x10001 = 256\*256+1

比65537更大的下一个素数应该是:

65536\*65536+1 = 0x1 0000 0001 = 4294967297

注: 身边的软硬件厂商基本都倾向于选取公钥指数e=65537=0x10001作为运算效率和安全的折衷选择.

5. 求私钥指数d

已知e=17与φ(n)= 3120互质，根据求模反元素的公式计算e的模反元素d：

令(e\*d)÷3120的余数等于1,

即

令整数k = (17\*d+1) ÷ 3120

其中d的取值范围为[1～3233] 要求1<=d<=n=3233

最终确定d的值

d = 2753, k = 15

17\*d = (3120\*k – 1)