비대칭 키 암호화 숙제

1. 파이썬의 cryptography 모듈에서 지원하는 rsa의 경우, key_size가 x 비트이고, 패딩을 위한 해시 알고리즘이 y 비트를 사용한다고 할 때, 암호화할 수 있는 최대 메시지의 길이는 x/8 - 2*y/8 - 2 바이트이다. 예를 들면, key_size가 2048비트이고 SHA256 해시 알고리즘을 패딩으로 사용할 경우, 메시지의 최대 길이는 190바이트이다. 이때 암호화할 파일의 크기가 190바이트를 넘어갈 경우, 강의 노트에 나온 방식으로 단순하게 암호화를 진행하면 오류가 발생한다.

   이를 해결하기 위하여 다음과 같은 암호화 정책을 사용할 수 있다.

   가. 길이가 긴 메시지는 AES를 이용하여 암호화한다. ← enc_msg
   나. 공개키를 이용하여 AES의 키(aes_key)를 암호화한다. ← enc_key
   다. enc_msg와 enc_key를 insecure channel로 수신자에게 전달한다.
   라. 수신자는 개인키를 이용하여 enc_key를 복호화한다. ← aes_key
   마. aes_key를 이용하여 enc_msg를 복호화하여 평문을 복원한다.

   위의 과정을 수행하는 파이썬 프로그램을 작성하라. 평문 메시지는 사용자 입력으로 받고, RSA의 공개키와 개인키는 첨부한 public_key.pem 파일과 private_key.pem 파일에 저장된 키를 이용한다. enc_msg의 복호화 결과를 출력하여, 사용자 입력 메시지와 동일함을 보이면 된다.

(다음 장에 계속)

2. 비트코인의 공개 키는 아래와 같은 SECP256K1 타원 곡선을 이용한다.

$Y^2 = (X^3 + 7) \% p$

p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F

개인키가 x라고 하면 공개 키는 x * G의 결과로 생성되는데, G는 타원 곡선상의 고정된 점으로 좌표는 다음과 같다.

G = (0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798,
0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)

256비트의 개인키를 random으로 생성한 후, 이에 대한 공개키를 계산하여 출력하는 프로그램을 작성하라. 생성된 개인키의 randomness를 강화하기 위하여 os.urandom()과 random.random(), 그리고 time.time()을 모두 적용한 문자열을 생성하고, 이를 hashlib.sha256() 함수를 이용하여 256비트의 난수를 생성한다. 생성된 난수의 크기가 SECP256K1 곡선의 p보다 작으면 개인키로 사용하고, 아니면 또 다른 난수를 다시 생성한다.

생성된 개인키에 대한 공개키를 만들기 위하여 강의 시간에 설명한 Extended Euclidian 알고리즘을 이용하여 곱셈의 역원을 계산하고, 더하기 연산은 강의 노트 21페이지의 직선 방정식을 이용하면 된다. 마지막으로 x * G의 곱하기 연산은 강의 노트 27페이지의 double-and-add 알고리즘을 이용하라. 테스트를 위한 개인키와 공개키의 조합은 다음과 같다.

개인키(16진수) = 0x771ab89947b6e39e1aaa7610085e5657e1eef2da7ccdf7af7d35b0413e661d38
개인키(10진수) = 53872441058844996679977158571737064850247033767869768697312274424220201917752

공개키(16진수) = (0xbd817926b132a6ef0e64dabd89424bfed3ba6c26cc667db3032fe0beffaac0e3,
0x33316b170d3204fe5ebf42a146361de23da9ba5242f1653635e6a9aad6f1bcf4)
공개키(10진수) = (85715887808040806675613841265983152055545362891430588938507562522056741011683,
23155269892248243197096053258448092991771802638930588277272349304738045607156)

참고로 https://paulmillr.com/ecc/ 사이트는 온라인으로 개인키를 입력받고, 그에 해당하는 공개키를 계산하여 출력한다.