2019 국가암호공모전 Ⅱ 분야 답안 제출 양식

소속 : 서울시립대학교 수학과　　대표자 이름 :

문제 04

답) I am Junyeong Lee of Anyang Gyeonggi-do. The reason why I am writing this message is because I want to leave my legacy to posterity so that I won't be forgotten. But somehow everything I know seems to be finally figured out by the NSA for now, so I can't leave a record anywhere. So I decided to write this message and encrypt it then erase the encryption key forever. With only the cipher-text alone, the NSA will not be able to figure out anything. The current elliptic curve crypto-technology would not be able to find out the plaintext. After 20 or 30 years, if quantum computers are built,this message would be revealed. Perhaps then I will nolonger be in the world. I appreciate you who at last found.this message. Perhaps you know the key is '71e3e7d3fac11617c282d57c4ab211e2' now. I am Junyeong Lee of Anyang Gyeonggi-do. I have organized and executed the world's biggest cyberattacks in the last decade. Let me start the story now.

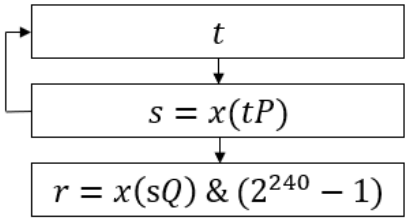
개요 :

1. 문제 분석

암호문과 평문의 일부가 주어졌다.
그리고 암호문을 생성한 알고리즘의 python 코드를 함께 제공받았다.

1-1. 코드 분석

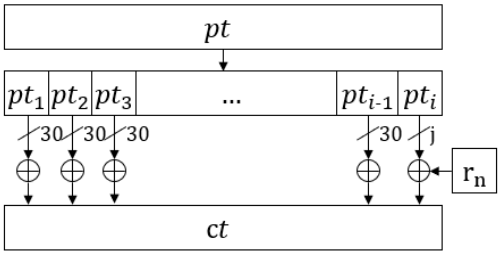
1-1-1. genStream 함수



< 그림 1. genStream 함수 >

P, Q 점들은 주어져있다.

1-1-2. encryption 함수



< 그림 2. encryption 함수 >

단, $0 \le j \le 30$이고 $n = 1, 2, ..., i$이다.

1-2. 취약점 발견하기

문제에서 주어진 타원곡선의 $order$는 p와 같다. 따라서 $Smart's\ Attack$[1]이 가능하다. 이 $Smart's\ Attack$을 이용하면, $P = s \bullet Q$ (P, Q는 타원곡선 위의 점, s는 상수)에서 P, Q가 주어져 있을 때, s를 쉽게 복구할 수 있다.

풀이 :

1. 평문 위치 찾아내기

주어진 일부 평문은 32바이트이다. $Smart's\ Attack$이 걸리는 속도를 고려하여, 30바이트 2바이트로 쪼개져 있는 경우, 1바이트 30바이트 1바이트로 쪼개져 있는 경우, 2바이트 30바이트로 쪼개져 있는 경우로 먼저 나누어볼 수 있었다.

21바이트 11바이트로 쪼개져 있는 경우를 생각하지 않은 이유는 일부 평문으로 평문이 끝나는 것은 문법에 어긋난다고 생각했기 때문이다.

1-1. 30바이트 2바이트로 쪼개져 있는 경우

주어진 암호문을 ct라 하자. ct는 941바이트이므로 30바이트씩 32번에 걸쳐 만들어졌을 것이다.

$ct = (ct[0], ct[1], ..., ct[31])$ 크기 32의 30바이트 단위벡터로 표현하고, 주어진 일부 평문을 p이라고 하고 상위 30바이트를 $p_1 = 'y\ figured\ out\ by\ the\ NSA\ \text{for}\ n'$라 하자.

① for $0 \le i \le 31$, $key[i] = ct[i] \oplus p_1$, ($pt[i] \oplus r_i = ct[i]$이므로 r_i를 찾기 위해 $key[i] = ct[i] \oplus p_1$를 한다.)

② for $0 \le j \le 0\text{x}f96e$, $k_i = key[i] + (j \ll 240)$

그림 1을 보면 r은 하위 30바이트만 사용되므로 버려진 상위 2바이트를 찾아내야 한다.

상위 2바이트만 생각하면 $0\text{x}ffff$이겠지만 문제에서 주어진 타원곡선의 $prime$를 생각하여 $0\text{x}f96e$까지만 하면 된다.

③ k_i에 대해 타원곡선에서 x좌표가 k_i인 점들을 찾아준다. 그 점들을 A_1, A_2라 하자. 존재하지 않는다면 다음 j로 넘어가자.

$prime$은 $4m+3$꼴이기 때문에, $y^2 = k_i^3 + ak_i + b \pmod p$에서 y의 해는 최대 2개이다. ($prime$을 p라 하자)

④ $A_m\ (m = 1, 2)$에 대해 $A_m = t_m Q$에서 $Smart's\ Attack$을 이용하여 t_m을 찾자.

⑤ $s_m = x(t_m P)$ 를 계산하자.

왜냐하면 t_m은 $i+1$에서 $seed$로 이용될 것이기 때문이다.

⑥ $k_{i+1,m} = (x(s_m Q)\&(2^{240}-1))$ 를 계산하자.

⑦ $pt[i+1]_m = k_{i+1,m} \oplus ct[i+1]$을 계산하여 상위 2바이트가 p의 하위 2바이트와 일치하는지 확인한다.

⑧ If 일치한다면, $pt[i+1]_m$이 아스키코드로 표현될 수 있는지 확인한다.

⑨ If 표현된다면 그때의 i, j 값을 저장한다.

이러한 방법을 통해 $i = 7, j = 60359$임을 알아낼 수 있었고 이어지는 평문이 $'ow,\ so\ I\ can't\ leave\ a\ record'$임을 알 수 있었다.

1-2. 1바이트 30바이트 1바이트로 쪼개져 있는 경우

주어진 일부 평문을 r이라고 하고 상위 30바이트를 $p_2 = '\ figured\ out\ by\ the\ NSA\ for\ no'$라 하자.

이후 과정은 1-1과 유사하지만, 평문을 확인하는 과정이 조금 다르다.

① for $1 \leqq i \leqq 31$, $key[i] = ct[i] \oplus p_2$

② for $0 \leqq j \leqq 0\mathrm{x}f96e$, $k_i = key[i] + (j \ll 240)$

③ k_i에 대해 타원곡선에서 x좌표가 k_i인 점들을 찾아준다. 그 점들을 A_1, A_2라 하자. 존재하지 않는다면 다음 j로 넘어가자.

④ $A_m\,(m = 1,2)$에 대해 $A_m = t_m Q$에서 $Smart's\ Attack$을 이용하여 t_m을 찾자.

⑤ $s_m = x(t_m P)$을 계산하자.

⑥ $k_{i+1,m} = (x(s_m Q)\&(2^{240}-1))$을 계산하자.

⑦ $pt[i+1]_m = k_{i+1,m} \oplus ct[i+1]$을 계산하여 상위 1바이트가 p의 하위 1바이트와 일치하는지 확인한다.

⑧ If 일치한다면, $pt[i+1]_m$이 아스키코드로 표현될 수 있는지 확인한다.

⑨ If 표현된다면 t_m에 대해 타원곡선에서 x좌표가 t_m인 점들을 찾아준다. 그 점들을 $B_{m,1}, B_{m,2}$라 하자. 존재하지 않는다면 다음 j로 넘어가자.

⑩ $B_{m,l}\,(m, l = 1,2)$에 대해 $A_{m,l} = s_{m,l} Q$에서 $Smart's\ Attack$을 이용하여 $s_{m,l}$을 찾자.

⑪ $k_{i-1,m,l} = (x(s_{m,l} Q)\&(2^{240}-1))$ 를 계산하자.

⑫ $pt[i-1]_{m,l} = k_{i-1,m,l} \oplus ct[i-1]$을 계산하여 하위 1바이트가 p의 상위 1바이트와 일치하는지 확인한다.

⑬ If 일치한다면, $pt[i-1]_{m,l}$이 아스키코드로 표현될 수 있는지 확인한다.

⑭ If 표현된다면 그때의 i, j 값을 저장한다.

1-3. 2바이트 30바이트로 쪼개져 있는 경우

주어진 일부 평문을 r이라고 하고 상위 30바이트를 $p_3 = '\ figured\ out\ by\ the\ NSA\ for\ now'$라 하자.

이후 과정은 1-1과 유사하지만, 평문을 확인하는 과정이 조금 다르다.

① for $1 \leqq i \leqq 31$, $key[i] = ct[i] \oplus p_3$

② for $0 \leq j \leq 0\text{x}f96e$, $k_i = key[i] + (j \ll 240)$

③ k_i에 대해 타원곡선에서 x좌표가 k_i인 점들을 찾아준다. 그 점들을 A_1, A_2라 하자. 존재하지 않는다면 다음 j로 넘어가자.

④ $A_m\,(m=1,2)$에 대해 $A_m = t_m Q$에서 $Smart's\ Attack$을 이용하여 t_m을 찾자.

⑤ t_m에 대해 타원곡선에서 x좌표가 t_m인 점들을 찾아준다. 그 점들을 $B_{m,1}, B_{m,2}$라 하자. 존재하지 않는다면 다음 j로 넘어가자.

⑥ $B_{m,l}\,(m,l=1,2)$에 대해 $A_{m,l} = s_{m,l}Q$에서 $Smart's\ Attack$을 이용하여 $s_{m,l}$을 찾자.

⑦ $k_{i-1,m,l} = (x(s_{m,l}Q)\&(2^{240}-1))$ 를 계산하자.

⑧ $pt[i-1]_{m,l} = k_{i-1,m,l} \oplus ct[i-1]$을 계산하여 하위 2바이트가 p의 상위 2바이트와 일치하는지 확인한다.

⑨ If 일치한다면, $pt[i-1]_{m,l}$이 아스키코드로 표현될 수 있는지 확인한다.

⑩ If 표현된다면 그때의 i, j 값을 저장한다.

2. key 찾기

1-1 경우에서 $i=7, j=60359$ 임을 알아냈으므로 그에 대해 k_7을 계산할 수 있다.

① 1-1-③, 1-1-④에서와 동일한 방법으로 A_1, A_2를 찾고 t_1, t_2.를 찾아준다. 이 때, A_1과 A_2의 x좌표는 같고 y좌표는 서로 $\bmod p$에서 덧셈의 역원이다.

A_1을 A라 하고 그에 대한 t_1을 t_7이라하자.

② for $1 \leq i \leq 7$, t_{8-i}에 대해 타원곡선에서 x좌표가 t_{8-i}인 점들을 찾아준다. 그 점들을 $B_{8-i,1}, B_{8-i,2}$라 하자.

③ $B_{8-i,1}$에 대해 $B_{8-i,1} = t_{7-i}P$에서 $Smart's\ Attack$을 이용하여 t_{7-i}를 찾자.

④ t_{7-i}에 대해 $k_{7-i} = (x(t_{7-i}Q)\&(2^{240}-1))$를 계산하자.

⑤ $pt[7-i] = k_{7-i} \oplus ct[7-i]$를 계산하여 $pt[7-i]$가 아스키코드로 표현되는지 확인한다.

⑥ If 표현되지 않는다면, $B_{8-i,2}$에 대해서도 동일하게 ③, ④, ⑤를 실시한다.

⑦ If 그래도 표현되지 않는다면, $i \geq 2$에서 $B_{9-i,2}$에 대해서 실시한 후, 다시 $B_{8-i,1}, B_{8-i,2}$에 대해 실시한다.

⑧ t_0에 대해 타원곡선에서 x좌표가 t_0인 점들을 찾아준다. 그 점들을 B_1, B_2라 하자.

⑨ B_1, B_2에 대해 $B_m = keyP$에서 $Smart's\ Attack$을 이용하여 key를 찾자.

즉, 하나씩 해보다가 아스키코드가 나오지 않는다면 전전 B의 y좌표를 바꿔주면 된다.

이러한 과정을 통해 $key = 0\text{x}71e3e7d3fac11f17c282d57c4ab211e2$임을 알 수 있었다.

3. 전체 평문 복구하기

복호화는 2에서 구한 key를 이용하여 암호화과정과 똑같이 진행하면 된다.

이를 통해 복구한 전체 평문은 "I am Junyeong Lee of Anyang Gyeonggi-do. The reason why I am writing this message is because I want to leave my legacy to posterity so that I won't be forgotten. But somehow everything I know seems to be finally figured out by the NSA for now, so I can't leave a record anywhere. So I decided to write this message and encrypt it then erase the encryption key forever. With only the cipher-text alone, the NSA will not be able to figure out anything. The current elliptic curve crypto-technology would not be able to find out the plaintext. After 20 or 30 years, if quantum computers are built,this message would be revealed. Perhaps then I will nolonger be in the world. I appreciate you who at last found.this message. Perhaps you know the key is '71e3e7d3fac11617c282d57c4ab211e2' now. I am Junyeong Lee of Anyang Gyeonggi-do. I have organized and executed the world's biggest cyberattacks in the last decade. Let me start the story now." 이다.

참고문헌 :

1) Nigel P. Smart, *The Discrete Logarithm Problem on Elliptic Curves of Trace One, HP Laboratories Bristol*, 1997
https://www.hpl.hp.com/techreports/97/HPL-97-128.pdf

2) Peter Novotney, *Weak Curves In Elliptic Curve Cryptography*, 2010
https://wstein.org/edu/2010/414/projects/novotney.pdf

1) Nigel P. Smart, *The Discrete Logarithm Problem on Elliptic Curves of Trace One, HP Laboratories Bristol* , 1997