

2023 암호분석경진대회

6번 문제

양자 내성 암호 중 isogeny 기반 암호는 유한체 위의 두 타원곡선 E, E' 사이의 isogeny를 연산하는 어려움에 기반을 두어, isogeny $\phi: E \rightarrow E'$ 에 대해 ϕ 를 비밀값으로 한다. 한편, Velu의 공식을 이용해 $\ker\phi$ 를 이용해 isogeny를 연산할 수 있으므로 $\ker\phi$ 도 비밀로 하며, 일반적인 구현에서는 isogeny를 저장하는 대신 $\ker\phi$ 를 저장한다.

다음은 isogeny 문제를 기반으로 하여 설계한 전자서명 알고리즘이다. Alice는 서명값 중 s 를 획득했다고 한다. 이를 이용해서 개인키를 복원하고, 해당 서명 알고리즘이 안전하지 않은 이유를 설명하시오.

(s 는 함께 제공된 2023pqc_s.txt 파일에서 확인할 수 있습니다.)

[Setup]

- $p = 2^{216}3^{137} - 1$, $E: y^2 = x^3 + 6x^2 + x \in F_p$
- $P_S, Q_S \in E[2^{216}]$, generator of $E[2^{216}]$, $P_R, Q_R \in E[3^{137}]$, generator of $E[3^{137}]$

[Key generation]

- 개인키 $n_S \in \{1, \dots, 2^{216} - 1\}$ 를 선택한 뒤 $S = P_S + n_S Q_S$ 연산한다
- $\langle S \rangle$ 를 커널로 하는 isogeny $\phi_S: E \rightarrow E_S$ 를 연산한다.
- 공개키: E_S , 개인키: n_S (ϕ_S)

[서명생성]

- INPUT : message
- Output : (e, s)
- 256개의 랜덤한 점 $R_0, \dots, R_{255} \in E[3^{137}]$ 에서 선택한다.
- $G_i = S + R_i$ 라 하고, 각 i 에 대해 $\phi(R_i)$ 를 연산한다.
- $\ker\beta_i = \langle \phi(R_i) \rangle$ 인 isogeny $\beta_i: E_S \rightarrow E_i$ 를 각 i 에 대해 연산한다.
- $r = (j(E_0) \parallel \dots \parallel j(E_{255}))$ 로 하고, $e = H(r \parallel m)$ 을 연산한다. 여기에서 j 는 j -invariant를 의미하고, H 는 SHA-256을 사용한다.
- $(b_0, \dots, b_{255}) = e$ 로 하여, b_i 를 0 또는 1이라 한다.
- $s = (K_0, \dots, K_{255})$ 를 다음과 같이 연산한다. $K_i = \begin{cases} G_i & b_i = 0 \\ \phi(R_i) & b_i = 1 \end{cases}$
- 서명은 (e, s) 이다.

[서명검증]

- INPUT : (e', s') , message
- Output : Valid, invalid
- 서명값 $e' = (b'_0, \dots, b'_{255})$ 와 $s = (K'_0, \dots, K'_{255})$ 를 이용해 다음과 같이 연산한다.
- 만약 $b'_i = 0$ 이면, $\ker\langle K'_i \rangle$ 인 isogeny $\alpha_i: E \rightarrow E'_i$ 를 연산한다.
- 만약 $b'_i = 1$ 이면, $\ker\langle K'_i \rangle$ 인 isogeny $\alpha_i: E_S \rightarrow E'_i$ 를 연산한다.
- $r = (j(E'_0), \dots, j(E'_{255}))$ 로 한 뒤, $c = H(r \parallel m)$ 을 연산한다. $c = e'$ 이면 valid, 그렇지 않으면 invalid를 출력한다.

[(참고)서명저장]

- 서명값 s 에 저장되는 타원곡선위의 점 K_i 는 x 좌표만 저장되며, 저장 시 특별한 압축 함수는 사용하지 않는다.
- K_i 는 x 좌표는 $a + bi$ 형태이며 a 가 448bit로 저장된 이후, b 가 448bit로 저장된다.
- Example) $GF(251^2)$ 에 정의된 K_1, K_2 의 x 좌표가 각각 $2+3i, 4+2i$ 라 하고 8bit에 저장된다 할 때 \Rightarrow $0x02, 0x03, 0x04, 0x02$ 로 저장

Point	좌표	값				
P_S	x	Re	00003CCFC5E1F050	030363E6920A0F7A	4C6C71E63DE63A0E	6475AF621995705F
		Im	7C84500CB2BB61E9	50E19EAB8661D25C	4A50ED279646CB48	
	y	Re	0001AD1C1CAE7840	EDDA6D8A924520F6	0E573D3B9DFAC6D1	89941CB22326D284
		Im	A8816CC4249410FE	80D68047D823C97D	705246F869E3EA50	
Q_S	x	Re	0001AB066B849495	82E3F66688452B92	55E72A017C45B148	D719D9A63CDB7BE6
		Im	F48C812E33B68161	D5AB3A0A36906F04	A6A6957E6F4FB2E0	
	y	Re	0000FD87F67EA576	CE97FF65BF9F4F76	88C4C752DCE9F8BD	2B36AD66E04249AA
		Im	F8337C01E6E4E1A8	44267BA1A1887B43	3729E1DD90C7DD2F	
$P_S - Q_S$	x	Re	0000C7461738340E	FCF09CE388F666EB	38F7F3AFD42DC0B6	64D9F461F31AA2ED
		Im	C6B4AB71BD42F4D7	C058E13F64B237EF	7DDD2ABC0DEB0C6C	
	y	Re	000025DE37157F50	D75D320DD0682AB4	A67E471586FBC2D3	1AA32E6957FA2B26
		Im	14C4CD40A1E27283	EAAF4272AE517847	197432E2D61C85F5	
$P_S - Q_S$	x	Re	0001D407B70B01E4	AEE172EDF491F4EF	32144F03F5E054CE	F9FDE5A35EFA3642
		Im	A11817905ED0D4F1	93F31124264924A5	F64EFE14B6EC97E5	
	y	Re	0000E7DEC8C32F50	A4E735A839DCDB89	FE0763A184C525F7	B7D0EBC0E84E9D83
		Im	E9AC53A572A25D19	E1464B509D97272A	E761657B4765B3D6	
P_R	x	Re	0000F37AB34BA0CE	AD94F43CDC50DE06	AD19C67CE4928346	E829CB92580DA84D
		Im	7C36506A2516696B	BE3AEB523AD7172A	6D239513C5FD2516	
	y	Re	000196CA2ED06A65	7E90A73543F3902C	208F410895B49CF8	4CD89BE9ED6E4EE7
		Im	E8DF90B05F3FDB8B	DFE489D1B3558E98	7013F9806036C5AC	
Q_R	x	Re	00007F65B303A50E	F1B4192237611E22	6A3D13384EF608A6	B117365A16E0EB51
		Im	12156F2012CB029C	819F3330F69BD5C7	3CCC9A1F1C06CD15	
	y	Re	0000749095AB8A36	C841FBF25A5671A6	7FDE5023131C73F0	EC6031C7E472DAE1
		Im	38FBED0A0BE63C67	06CD893EF88D32CC	766EC67EC056ED33	
$P_R - Q_R$	x	Re	00008664865EA7D8	16F03B31E223C26D	406A2C6CD0C3D667	466056AAE85895EC
		Im	37368BFC009DFAFC	B3D97E639F65E9E4	5F46573B0637B7A9	
	y	Re	00006AE515593E73	976091978DFBD70B	DA0DD6BCAEEBFDD4	FB1E748DDD9ED3FD
		Im	CF679726C67A3B2C	C12B39805B32B612	E058A4280764443B	
Q_R	x	Re	00012E84D7652558	E694BF84C1FBDAAF	99B83B4266C32EC6	5B10457BCAF94C63
		Im	EB063681E8B1E739	8C0B241C19B9665F	DB9E1406DA3D3846	
	y	Re	0			
		Im	0000EBAAA6C73127	1673BEECE467FD5E	D9CC29AB564BDED7	BDEAA86DD1E0FDDF
$P_R - Q_R$	x	Re	399EDCC9B49C829E	F53C7D7A35C3A074	5D73C424FB4A5FD2	
		Im	0001CD28597256D4	FFE7E002E8787075	2A8F8A64A1CC78B5	A2122074783F51B4
	y	Re	FDE90E89C48ED91A	8F4A0CCBACBFA7F5	1A89CE518A52B76C	
		Im	000147073290D78D	D0CC8420B1188187	D1A49DBFA24F26AA	D46B2D9BB547DBB6
E_S (공개키)	A'	Re	0000DA7A98EA2646	9B843EBF8D1EE0F0	0E6786E680AC535F	5FF26D25819549C9
		Im	59497D8E8FB14B1B	F6764BD27BAE970D	0791AF091E344F22	
	A'	Re	000048704FEC03D0	5B06D8A8197DF08D	4946E465099F31B7	5C63A865A23CA2AD
		Im	41A74F05074E9DC3	F45C5A26F741A0EA	1F3C2E6CDA0BB344	
A'	Re	0000BC39A8C22AFD	CAC43EFDD3AB05B4	5AF0A795D823CD1E	C0931D386BFDE2D4	
	Im	77DFFFBF2C846346	0DE0586510E99F24	323AB8E54BD0026B		
A'	Re	0000045E901E3BAA	12BA1A2D0A37634D	EF74A6791039D723	962496EB9C4C368F	
	Im	D50BD06BC7D7EF0B	2582ADF73577537B	DAA9A064C9AB0DA5		