프로젝트 #5 (팀)

B08조 – 윤동현, 홍석민, 윤재현, 노건웅

함수에 대한 설명

- `int rsaes_oaep_encrypt(const void *m, size_t mLen, const void *label, const void *e, const void *n, void *c, int sha2_ndx)`: 길이가 mLen 바이트인 메시지 m을 공개키 (e,n)을 사용하여 암호화한 결과를 c에 저장합니다. 과정은 다음과 같습니다. 첫번째로, label을 hash한 결과를 메시지와 함께 주어진 형식에 맞추어 DB에 저장합니다. 두번째로 randomly하게 생성된 seed와 만들어진 DB를 MGF 함수를 이용하여 maskedSeed, maskedDB를 생성한 후 EM에 저장합니다. 이 때 EM은 RSA 키의 길이를 넘지 않도록 맨 앞에 00 바이트를 붙여 줍니다. 마지막으로 생성된 EM을 공개키 (e, n)으로 암호화한 뒤 c에 저장합니다.
- `int rsaes_oaep_decrypt(void *m, size_t *mLen, const void *label, const void *d, const void *n, const void *c, int sha2_ndx)`: 암호문 c를 개인키 (d,n)을 사용하여 원본 메시지 m과 길이 mLen을 회복합니다. label과 sha2_ndx는 암호화할 때 사용한 것과 일치해야 합니다. 성공하면 0, 그렇지 않으면 오류 코드를 넘겨 줍니다. 인자로 넘겨받은 c를 개인키를 사용한 rsa_cipher를 통해 복호화하여 EM을 얻어냅니다. 그 후 "EM = 00 || maskedSeed || maskedDB"의 구조로 분리한 후 MGF1 함수를 통해 seed와 DB를 분리합니다. 그리고 나서 DB에서 message를 얻어냅니다.
- `int rsassa_pss_sign(const void *m, size_t mLen, const void *d, const void *n, void *s, int sha2_ndx)`: 메시지 길이인 `mLen`이 hash의 최대 입력 길이보다 작은 지 확인 후 메시지를 hash합니다. 다음으로 "m' = (0x)00 00 00 00 00 00 00 00 || mHash || salt"으로 m'을 얻어 냅니다. "DB = PS || 0x01 || salt" 인 DB를 얻어냅니다. m'를 해시한 h을 얻은 후 MGF1 함수에 h를 넣어 얻은 값을 DB와 XOR연산을 합니다. 이때 나온 값을 maskedDB라고 했을 때 "EM = maskedDB || h || 0xbc"인 길이가 RSAKEYSIZE 비트인 EM을 얻습니다. 이때 맨 왼쪽 마지막 비트가 1이면 강제로 0으로 치환해줍니다. 마지막으로 EM을 (d,n)으로 암호화하여 값을 s에 저장합니다.
- `int rsassa_pss_verify(const void *m, size_t mLen, const void *e, const void *n, const void *s, int sha2_ndx)`: 길이가 mLen 바이트인 메시지 m에 대한 서명이 s가 맞는지 공개키 (e,n)으로 검증합니다. sign 함수에서 서명하는 메시지(M)가 EM으로 변한 되는 과정의 역순이라고 생각하면 편합니다. mHash와 Hash를 이용해 MGF를 만들고 DBmask 와 maskedDB를 xor 연산을 통해서 DB값을 알아냅니다. M' = 0x00(8byte)|| mHash || salt로 만든 후에 다시 hash를 통해서 H'를 생성 후 H 값과 H' 값 비교를 통해 인증을 확인합니다.
- `unsigned char* MGF1(unsigned char* mgfSeed, size_t mgfSeedLen, size_t maskLen, void (*fptr)(unsigned char*, size_t, unsigned char*), size_t hLen)`: 길이가 mgfSeedLen + 4인 temp에 mgfSeed를 넣습니다. 이후 counter가 0부터 (maskLen / hLen) – 1의 반올림 값까지인 반복문을 통해 temp에 있는 4 octets에 counter을 넣습니다. 다음으로 sha 함수를 가리키고 있는 fptr에 temp을 넣고 hCounter를 반환합니다. 이후 반환할 값인 mgf에 "mgf = mgf || hCounter"를 합니다. 반복문이 끝나면 mgf을 반환합니다.
- `void* select_sha(int sha_nda, size_t* hLen)`: sha_nda의 값을 확인 후 값에 해당되는 sha의 반환 길이를 hLen에 저장하고 sha함수를 반환합니다.

컴파일

```
❯ make
gcc -Wall -O3 -c test.c
gcc -Wall -O3 -c pkcs.c
gcc -Wall -O3 -c sha2.c
gcc -o test test.o pkcs.o sha2.o -lgmp
```

```
❯ ./test
e = 01ce1c638e426180850ef9f3c74191811ed6d389de6166b89d35db6dbff4826ac6c3a21d7d7d810dfd81ab6a2f4346d2073678bad9e08ce7c0b445b9e714e66110e50bcb7222dc8a85494d58189161a68b1dd3e5
8f8257aa2986cf46e47fc73c882fad57380c16cc7fcc3e5c635ec9da0dd511fc80d488ad7bdec012a15170461e5879953a8efdfa6732b2cef42cb57d2f9eccb55ca7b55b49cfa2ad1f3a10f30b2961ba45c5a305906e
3bfcfe47e99b5e26b87457089d09821279b3aadd297ac7ce3531ce742855d324ff9b4e31c216bd1c399764bd3e22ada95a7fb651415e7968f24019402bc504cc0538da9fd3151d7ad4383f0980557ea868eeb34cff3d
d = 0e8cbe47dfdddd7068dc933fb8cf8b3d600d768a6d0a85b961affc6fe3c5fd1e73a889b1cb4b1f1b1518775f6eb3225bee7c12077e940a7720519f7b5cf21260ba64642d2560cfe7f4dc1ede43560bc7ad89d0d5
7e12df225856c24bd06f9d98a158db4db686a1a7bea18647a53907a5f20bde9f4a63f22594cd9f7f0228deb93b2ede495b503e5209ccf02f03edb88392a7236be9602e2b8462b84bf1ef1c13fae181bab13a3a539810
b69a999dda07d87b30eda87b763dac70176c3cc1e52df142c338a6df4cc2263911237c7fedcf1c199e59a715ae7095cdb3d5254b2933669890e0967fdecd053ac3600f8aa31fc5a346d29b90bed5287bcdc30b98e42d
n = 9dd14bf48a2099e8e232f53f410b0633feb79622da90933dc29c7f05c8a26eb36b974dc4f846d8e81c7927fb5ab878779955987de5ddb5484951487b15502b533cbce9dbcdfcbf5163cc2063a84a0efddf278542
aed209361eb8bf968777bf451a17f69fd6572448e356763d38441eff39ea708f648a9659573982e186b4f2601b7495892d87af07e2777659511b4faa85af09f29c030fe8d48d83c853878d7b1af03d749fd653402868
4705385502cf851ba0b5114b2e90fe12ef37769ed881edd25b96f10d4db2e2b81aff17b95063f7ed605ee515b8b585772b0fb629ac0ebcc175b3e15aced8ed314838cd5694ebf6c633805f591d0c0b92bcfdc2c2e92d
---
c = 51d120911c1dbe8d3f21a1de6c3d6d9e61dc31e78f69dcf117fe63b6e7040fcbd48db640d682c56ead1f71ca48875aa23233ce333af4896fa4628090992c524f38ca58ea105626d86889e4ea17af1d7460c95ac9
a3bbc2b7e1d741fb6d34fa1c2052f1a3e87a54a8783d6bdbb7db6e24e2b49a445d2948d3c6067071c0eb92136cc21994445c78d15d76dbb9eac7b80f09f07379355ca6e70c6df0a3d1e70f0c5d478064610ad36ce562
f6ee3ee15c88a00763ad9f5d2165faf756b68d5e69bfb61e3533b52eeecd7b09707ad24acedac9ece7a2aa533b325ba9b4e4a573c8356385054cf9e3d085a57368d52befa6a9fe4c4ef47885d6cb7fb78b1963dce6bc
m = 73616d706c65206461746100
msg = sample data, len = 12 -- PASSED
---
c = 92217c88ee32097415006d884772138f881d719ebeef8bc5917ccf79c95499e2a997e12f50675a09802d783e2530439f06b033d8a300ba6856694087c8dfb4aeda629a8ba1870de68e2ab2775c1011a672311f79
5493a7a5d450857d9193487764404cc7ee446c8dfbddab36fd9da8c083b093008ac9e83e25adce55aa7e04ecd23aec1e1368cdd371b8c54e6720ffea277dff1734e2e45a3d223efee121fa14c6435a13e4e1d645797f
14435424bfc4909a029bfc7972c5ceef1b87e5534af739d6b61ace555635f602e896b8ffa946550d02501f3fdf8278e280cd65e490d36a20dde241e9cafe5d6866a4a2370a7078c3f5e21e55635e5a7cd8d4ca5aaa9a
m = 6d617820646174610 06c6162656c006d61782b206461746100456e6372797074696f6e204572726f723a2025642c206d65737361676520697320746f6f206c6f6e67202d2d205041535345440a2d2d2d0a004465
6372797074696f6e204572726f723a206d657373616765206c656e67746820257a75206973206e6f74207a65726f202d2d204641494c45440a006b6f7265616e20706f6574000a6d7367203d2025732c206c656e203d
20257a75202d2d205041535345440a0045696e73
msg = max data, len = 190 -- PASSED
---
Encryption Error: 2, message is too long -- PASSED
---
c = 0748f351403dacc2e25ac173e81fb52cc41d083d71c123e3ae2de763936f089ec8be1ef094351b13cfbe0f39313bbbcbe4f55776912f3c57a22da61db869fdfdccb63bde3b0e01336de2c8dc2cfc2592c46fcbf4
de7b2cc5ac75b2fe56cadbd8e33deb99ac6ee26bab970d131ed44ab9e5d91ef93b90dcb77f0e24799faed5c8743bd40c89dfc787ae285bdaf0ffd368afa44c8ac92ba2f073807bed50e065f3c40e164c3cc69283699f
45241b184e0291f07facd228fc4a5d1e3b9d6cb452518f48ff1539d0e6eee4bc02e5db05dfba0afe9081e604a3dba40ee7d20391d38e2f6c3a9c1fd59ffc626107c82621e4fe65f8f68e8823a3faefffef76f30a8ba5
Empty message -- PASSED
---
m = ec9ca4eb8f99eca3bc
msg = 윤동주, len = 9 -- PASSED
msg = 배움은 경험에서 얻는다. 경험하지 않은 것은 정보에 불과하다. - 아인슈타인 -- PASSED
---
RSAES-OAEP Random Testing.....................................................................................No error found! -- PASSED
```

```
---
s = 05568806aed5cac680cde4a15a9954d6a15f74ec1182834d131ecbc015064564a84e3836a05ae81d8eef40cb471fc4f202cfecac1fea03e5f1f24a68e6f26f39d24e7775d579524f0462c3b7fdfca099edb66398
7eafe811943fb31dda378c8cda21337c11cb8b52b92a75808fbbeb274ed6483636f09a7f1d81ef2a28e74e0e37ea0e7f84933a680e16e24c47561aae533d8609b47b321abcbd8486313b6552dea959ee3cbc4c325d9e
432d06dd0aed64f5b84f3d0b83fb00d74539727e1365a847b7521facb4287b44f78c46956502e4324844cf76a4755e7c51a23db0140cb0731123d4ea5b7c4cbb9d8241a97abb54571d5c8b535e9d48958e2f3caac4fb
Valid Signature! -- PASSED
---
Hash Input Error: 2 -- PASSED
---
s = 178bb00f94e2ca196cf95245b095015017c2e40853b178a7bcff96a8a64815d869b5fcf6ed5c38e2cabaf325dfa2f1a51b459f847326a305f857757694530727805204a685b055ffa4cf1772da6128ab4a115f0c
5a45688b8cc73e06268ee6cb52bf88b3d4d693ab473fe99e361244259895658c009039c05f13e794855a0889320c38f5ed7738f17e18cd60a92b7d248686c191861ed53705e9758fab8c237a427166f935792a79387e
ec89157d36658e5bc2cb7e1703753d7df6b4e86ebfbd04167d678a3f32243bc985d12f09cdab92f4fbd609dc7f556d346f80d9bf883817461f7c49a0f41f0e14fb815e96d9c1e997df450b18a7b4de305cf91f83032e
Verification Error: 5, invalid signature -- PASSED
---
s = 26bf4429a000c4b0e6c32a5ed435b2aa4962fcf19098dfcbec31efc42fe1b3621d4329b10ca48adc8709f2a44be9b733b8bf9f9ff29fb28d91b2119d9d70b2e3859eddca5165d4192193064280d7634930cee580
d19dbf209b71956cb10762bb90295df9cade0f422039ca4efd5f99ba33f72554bf3bbc414c0ac6037a23980614f5f1be78356a70f25db06f551009fda30c6db63f2e3b3277eadce679026efefc335fb714462ca1e879
f3f222b27c2a5efe6278c264a8325232b61f9f7583f7b733fcf6819453e9af9081847d12722ce75e0eedd49859d36d85b5bde801261e6ab58e3385a21ee20a0d748525cfcfa734aaa5b80459b8ee53efe8266060a2ba
Verification Error: 8 -- PASSED
---
s = 00007e1fd623b6bddadf447b70772bec3fdfb1562e52bcf4fcc5e8f3aa0bd38a5488f9d8d8c9177e76a6172cad13e451e26af0d53cc78149218cee941a8952e88e21c78811afd7354fe1eb8e2178131c18d6d7ee
4df8746d90a90562f796c3cbab29dca7cbc9613341a5c73aea74bba98bb36d067196d22b47a57a56c7a3be501b0a0d7170fc2316347a39075c33316186f441fabd007438478aa03e967a221da1540ef815459618010d
7a1d0986b56ad0389dc47a00cd006f3a488dd794afb355f6ddddda81e7c2ae4d4246affa6e876bd7285c1c21cadef94848856822332c13764a3fabb0d8189399d8dd7790e80854e7db279f4e68c69769c75af22180dbd
Verification Error: 8 -- PASSED
---
Compatible Signature Verification! -- PASSED
---
RSASSA-PSS Random Testing.....................................................................................No error found! -- PASSED
CPU 사용시간 = 81.0592초
```

실행 결과물의 주요 장면과 설명

(1) RSAES-OAEP

```
e = 01ce1c638e426180850ef9f3c74191811ed6d389de6166b89d35db6dbff4826ac6c3a21d7d7d810dfd81ab6a2f4346d2073678bad9e08ce7c0b445b9e714e66110e50bcb7222dc8a85494d58189161a68b1dd3e5
8f8257aa2986cf46e47fc73c882fad57380c16cc7fcc3e5c635ec9da0dd511fc80d488ad7bdec012a15170461e5879953a8efdfa6732b2cef42cb57d2f9eccb55ca7b55b49cfa2ad1f3a10f30b2961ba45c5a305906e
3bfcfe47e99b5e26b87457089d09821279b3aadd297ac7ce3531ce742855d324ff9b4e31c216bd1c399764bd3e22ada95a7fb651415e7968f24019402bc504cc0538da9fd3151d7ad4383f0980557ea868eeb34cff3d
d = 0e8cbe47dfdddd7068dc933fb8cf8b3d600d768a6d0a85b961affc6fe3c5fd1e73a889b1cb4b1f1b1518775f6eb3225bee7c12077e940a7720519f7b5cf21260ba64642d2560cfe7f4dc1ede43560bc7ad89d0d5
7e12df225856c24bd06f9d98a158db4db686a1a7bea18647a53907a5f20bde9f4a63f22594cd9f7f0228deb93b2ede495b503e5209ccf02f03edb88392a7236be9602e2b8462b84bf1ef1c13fae181bab13a3a539810
b69a999dda07d87b30eda87b763dac70176c3cc1e52df142c338a6df4cc2263911237c7fedcf1c199e59a715ae7095cdb3d5254b2933669890e0967fdecd053ac3600f8aa31fc5a346d29b90bed5287bcdc30b98e42d
n = 9dd14bf48a2099e8e232f53f410b0633feb79622da90933dc29c7f05c8a26eb36b974dc4f846d8e81c7927fb5ab878779955987de5ddb5484951487b15502b533cbce9dbcdfcbf5163cc2063a84a0efddf278542
aed209361eb8bf968777bf451a17f69fd6572448e356763d38441eff39ea708f648a9659573982e186b4f2601b7495892d87af07e2777659511b4faa85af09f29c030fe8d48d83c853878d7b1af03d749fd653402868
4705385502cf851ba0b5114b2e90fe12ef37769ed881edd25b96f10d4db2e2b81aff17b95063f7ed605ee515b8b585772b0fb629ac0ebcc175b3e15aced8ed314838cd5694ebf6c633805f591d0c0b92bcfdc2c2e92d
```

1. RSA 암·복호화에 사용할 e, d, n을 생성합니다.

```
c = 51d120911c1dbe8d3f21a1de6c3d6d9e61dc31e78f69dcf117fe63b6e7040fcbd48db640d682c56ead1f71ca48875aa23233ce333af4896fa4628090992c524f38ca58ea105626d86889e4ea17af1d7460c95ac9
a3bbc2b7e1d741fb6d34fa1c2052f1a3e87a54a8783d6bdbb7db6e24e2b49a445d2948d3c6067071c0eb92136cc21994445c78d15d76dbb9eac7b80f09f07379355ca6e70c6df0a3d1e70f0c5d478064610ad36ce562
f6ee3ee15c88a00763ad9f5d2165faf756b68d5e69bfb61e3533b52eeecd7b09707ad24acedac9ece7a2aa533b325ba9b4e4a573c8356385054cf9e3d085a57368d52befa6a9fe4c4ef47885d6cb7fb78b1963dce6bc
m = 73616d706c65206461746100
msg = sample data, len = 12 -- PASSED
```

2. 메시지로 “sample data”가 주어졌을 때 RSAES-OAEP를 이용하여 암·복호화가 되는지 확인합니다.

```
c = 92217c88ee32097415006d884772138f881d719ebeef8bc5917ccf79c95499e2a997e12f50675a09802d783e2530439f06b033d8a300ba6856694087c8dfb4aeda629a8ba1870de68e2ab2775c1011a672311f79
5493a7a5d450857d9193487764404cc7ee446c8dfbddab36fd9da8c083b093008ac9e83e25adce55aa7e04ecd23aec1e1368cdd371b8c54e6720ffea277dff1734e2e45a3d223efee121fa14c6435a13e4e1d645797f
14435424bfc4909a029bfc7972c5ceef1b87e5534af739d6b61ace555635f602e896b8ffa946550d02501f3fdf8278e280cd65e490d36a20dde241e9cafe5d6866a4a2370a7078c3f5e21e55635e5a7cd8d4ca5aaa9a
m = 6d617820646174610 6c6162656c006d61782b206461746100456e6372797074696f6e204572726f723a2025642c206d6573736167652069732073207 46f6206c6f6e67202d2d205041535345440a2d2d2d0a004465
6372797074696f6e204572726f723a206d657373676165206c656e67746820257a75206973206e6f74207a65726f202d2d204641494c45440a006b6f7265616e20706f6574000a6d7367203d2025732c206c656e203d
20257a75202d2d205041535345440a0045696e73
msg = max data, len = 190 -- PASSED
```

3. 길이가 190바이트인 메시지도 RSAES-OAEP를 이용하여 암·복호화가 되는지 확인합니다.

```
Encryption Error: 2, message is too long -- PASSED
```

4. RSAE- OAEP가 허용하는 길이를 초과한 메시지가 들어왔을 때 암호화할 수 없다는 ERROR Message를 내보냅니다.

```
c = 0748f351403dacc2e25ac173e81fb52cc41d083d71c123e3ae2de763936f089ec8be1ef094351b13cfbe0f39313bbbcbe4f55776912f3c57a22da61db869fdfdccb63bde3b0e01336de2c8dc2cfc2592c46fcbf4
de7b2cc5ac75b2fe56cadbd8e33deb99ac6ee26bab970d131ed44ab9e5d91ef93b90dcb77f0e24799faed5c8743bd40c89dfc787ae285bdaf0ffd368afa44c8ac92ba2f073807bed50e065f3c40e164c3cc69283699f
45241b184e0291f07facd228fc4a5d1e3b9d6cb452518f48ff1539d0e6eee4bc02e5db05dfba0afe9081e604a3dba40ee7d20391d38e2f6c3a9c1fd59ffc626107c82621e4fe65f8f68e8823a3faefffef76f30a8ba5
Empty message -- PASSED
```

5. RSAES-OAEP를 이용하여 빈 메시지 암·복호화가 되는지 확인합니다.

```
m = ec9ca4eb8f99eca3bc
msg = 윤동주, len = 9 -- PASSED
msg = 배움은 경험에서 얻는다. 경험하지 않은 것은 정보에 불과하다. - 아인슈타인 -- PASSED
```

6. 암호문을 RSAES-OAEP로 복호화하여 사전에 정해놓은 메시지와 동일한지 확인합니다.

```
RSAES-OAEP Random Testing.....................................................................................No error found! -- PASSED
```

7. RSAES-OAEP 무작위 검사합니다.

(2) RSASSA-PSS

```
s = 05568806aed5cac680cde4a15a9954d6a15f74ec1182834d131ecbc015064564a84e3836a05ae81d8eef40cb471fc4f202cfecac1fea03e5f1f24a68e6f26f39d24e7775d579524f0462c3b7fdfca099edb66398
7eafe811943fb31dda378c8cda21337c11cb8b52b92a75808fbbeb274ed6483636f09a7f1d81ef2a28e74e0e37ea0e7f84933a680e16e24c47561aae533d8609b47b321abcbd8486313b6552dea959ee3cbc4c325d9e
432d06dd0aed64f5b84f3d0b83fb00d74539727e1365a847b7521facb4287b44f78c46956502e4324844cf76a4755e7c51a23db0140cb0731123d4ea5b7c4cbb9d8241a97abb54571d5c8b535e9d48958e2f3caac4fb
Valid Signature! -- PASSED
```

1. 문자열 "sample"을 개인키로 서명하고 공개키로 검증합니다.

```
Hash Input Error: 2 -- PASSED
```

2. 해시함수가 허용하는 메시지의 최대 길이를 초과한 경우를 시험한다.

```
s = 178bb00f94e2ca196cf95245b095015017c2e40853b178a7bcff96a8a64815d869b5fcf6ed5c38e2cabaf325dfa2f1a51b459f847326a305f8577576945307278052O4a685b055ffa4cf1772da6128ab4a115f0c
5a45688b8cc73e06268ee6cb52bf88b3d4d693ab473fe99e3612442598956580009039c05f13e794855a0889320c38f5ed7738f17e18cd60a92b7d248686c191861ed53705e9758fab8c237a427166f935792a79387e
ec89157d36658e5bc2cb7e1703753d7df6b4e86ebfbd04167d678a3f32243bc985d12f09cdab92f4fbd609dc7f556d346f80d9bf883817461f7c49a0f41f0e14fb815e96d9c1e997df450b18a7b4de305cf91f83032e
Verification Error: 5, invalid signature -- PASSED
```

3. 서명은 문자열 "invalid sample"에 하고 검증은 "invalid_sample"에 합니다.

```
s = 26bf4429a000c4b0e6c32a5ed435b2aa4962fcf19098dfcbec31efc42fe1b3621d4329b10ca48adc8709f2a44be9b733b8bf9f9ff29fb28d91b2119d9d70b2e3859eddca5165d4192193064280d7634930cee580
d19dbf209b71956cb10762bb90295df9cade0f422039ca4efd5f99ba33f72554bf3bbc414c0ac6037a23980614f5f1be78356a70f25db06f551009fda30c6db63f2e3b3277eadce679026efefc335fb714462ca1e879
f3f222b27c2a5efe6278c264a8325232b61f9f7583f7b733fcf6819453e9af9081847d12722ce75e0eedd49859d36d85b5bde801261e6ab58e3385a21ee20a0d748525cfcfa734aaa5b80459b8ee53efe8266060a2ba
Verification Error: 8 -- PASSED
```

4. 올바르지 않은 검증키를 사용해서 서명 검증을 시도합니다.

```
s = 00007e1fd623b6bddadf447b70772bec3fdfb1562e52bcf4fcc5e8f3aa0bd38a5488f9d8d8c9177e76a6172cad13e451e26af0d53cc78149218cee941a8952e88e21c78811afd7354fe1eb8e2178131c18d6d7ee
4df8746d90a90562f796c3cbab29dca7cbc9613341a5c73aea74bba98bb36d067196d22b47a57a56c7a3be501b0a0d7170fc2316347a39075c33316186f441fabd007438478aa03e967a221da1540ef815459618010d
7a1d0986b56ad0389dc47a00cd006f3a488dd794afb355f6ddddda81e7c2ae4d4246affa6e876bd7285c1c21cadef94848856822332c13764a3fabb0d8189399d8dd7790e80854e7db279f4e68c69769c75af22180dbd
Verification Error: 8 -- PASSED
```

5. 서명 값을 생성한 후 앞부분 2바이트를 0으로 바꾼 후 검증을 시도합니다.

```
Compatible Signature Verification! -- PASSED
```

6. 시인의 개인키로 서명된 시를 시인의 공개키로 검증합니다. RSA 키의 길이는 2048비트이고, SHA256 해시함수를 사용해야합니다.

```
RSASSA-PSS Random Testing.....................................................................................No error found! -- PASSED
```

7. RSASSA-PSS SIGN과 VERIFY을 무작위 검사 합니다.

소감 및 문제점

노건웅: RSAES-OAEP ENCRYPT 과정에서 가장 어려웠던 점은 공식 문서를 읽고 각 과정을 어떻게 구현해야하는 지 이해하는 것이었습니다. 혼자만의 힘으로 부족하여 RSAES-OAEP를 함께 구현하는 팀원과 같이 쉬는 시간마다 모여 문서를 이해하고 해석하는 것에 시간을 많이 쏟았습니다. 공식 문서를 이해한 뒤 구현하는 것은 문서에 나와있는 과정을 그대로 코드로 옮기는 것이기 때문에 어렵지 않았습니다. 각 과정마다 얼마 만큼의 메모리를 할당해야 하는지에 주의해서 코드로 구현하였습니다.

윤재현: 시험공부를 하면서 본 RSAES-OAEP 구조를 복습하는 기회를 가지게 되어서 너무 좋았습니다. 처음엔 그림을 보고 “이게 뭐지?”라고 생각했지만, 막상 팀 내부에서 분량을 나누고, 서로 개념 상으로 부족한 점들을 질문을 통해서 보충하고, 서로 협동을 하니 순조롭게 진행이 되어서 매우 즐거웠습니다. 정말 오랜만에 또 포인터를 사용해보니 힘들고, 재밌고, 뿌듯한 결과물이 나온 것 같아서 좋았습니다.

홍석민: RSAES-OAEP, RSASSA-PSS 두가지 구현된 것을 보면서 다르게 구성되나 구조적으로 봤을 때에는 비슷하는 것을 느낄 수 있었습니다. 팀원끼리 코드를 제작하면서 여러가지 문제점에 대해서 토론과 의견을 나누면서 해결하는 과정에서 협업에 대한 장점을 느낄 수 있었습니다.

윤동현: 팀프로젝트에서 서명부분에 대한 인증 함수(RSASSA-PSS VERIFY) 부분을 맡아서 팀프로젝트를 진행하게 되었습니다. RSASSA-PSS SIGN 함수와 RSASSA-PSS VERIFY 함수에 대한 문서에 대해 정리를 해보니 VERIFY 함수는 SIGN 함수가 진행하는 과정의 역행과 비슷하다고 느꼈습니다. 팀프로젝트를 진행하는 동안 제가 부족했던 부분을 많이 느껴서 배울 점이 많았다고 느꼈습니다.