2018 암호분석경진대회 : 6번 문제 답안

1) Premaster Secret 값을 구하라.

0303f5230bdb21aa5272c304c343c5b1712ed40845165b020542e9f9e43c7c58b6c4242c1eef8d94732bf980e19bcaed

2) Session Hash 값을 구하라.

bcebc78de9c67da8ade8f46e0799afcf206285522af1e7ad89d3a17ab1cd04fe

3) Master Secret 값을 구하라.

6db26c792fcdef2290ba4892bf6ad5ab0e43ae319cb5c993be7775da3ceefd2858db22291cddf63ada8eaf1599de49d9

4) Key Block을 구하라.

Key block = 0x3b82b13d1a0a5f51094005ca0dc2586d98b69ef4589b5109229fcdc8b161de05(client_MAC_key)
+ 0xf3aa5489548758c49ab9d851d5594e638c6eafeb07a728ca1b212b9d53e26475(server_MAC_key)
+ 0x647592711ae5e26f140c45d80323e922(client_enc_key)
+ 0xd6b2eb475c92649e9c977475eec403a7(server_enc_key)
+ 0x7356e0948a0544d5d8bfc01441a4f3d6(client_IV)
+ 0x82a5f1a84453659c7c9df430eec139a1(server_IV)

5) 암호문 C를 복호화 하여 P를 구하고, Character Array로 출력한 문장을 구하라.

Congratulations! You solved this problem.

먼저 문제에서 주어진대로 Server(220.149.25.211)와 Client(192.168.0.7)이 통신을 맺는 순간을 확인하였습니다. Wireshark를 사용하여 어떤 Cipher Suite를 사용하는지 우선적으로 확인하였습니다.

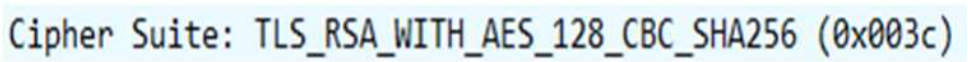
아래 있는 통신기록은 Client가 Server에게 보내는 패킷입니다.

13615	79.282222	192.168.0.7	220.149.25.211	TLSv1.2	234	Client Hello
14300	85.969006	192.168.0.7	220.149.25.211	TLSv1.2	234	Client Hello
14308	85.981368	192.168.0.7	220.149.25.211	TLSv1.2	234	Client Hello
14311	85.988785	192.168.0.7	220.149.25.211	TLSv1.2	412	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14314	85.994661	192.168.0.7	220.149.25.211	TLSv1.2	395	Application Data

아래 있는 통신기록은 Server가 Client에게 보내는 패킷입니다.

13616	79.286422	220.149.25.211	192.168.0.7	TLSv1.2	1060	Server Hello, Certificate, Server Hello Done
14301	85.971266	220.149.25.211	192.168.0.7	TLSv1.2	1060	Server Hello, Certificate, Server Hello Done
14309	85.984734	220.149.25.211	192.168.0.7	TLSv1.2	1060	Server Hello, Certificate, Server Hello Done
14312	85.993997	220.149.25.211	192.168.0.7	TLSv1.2	145	Change Cipher Spec, Encrypted Handshake Message
14323	86.036525	220.149.25.211	192.168.0.7	TLSv1.2	571	Application Data

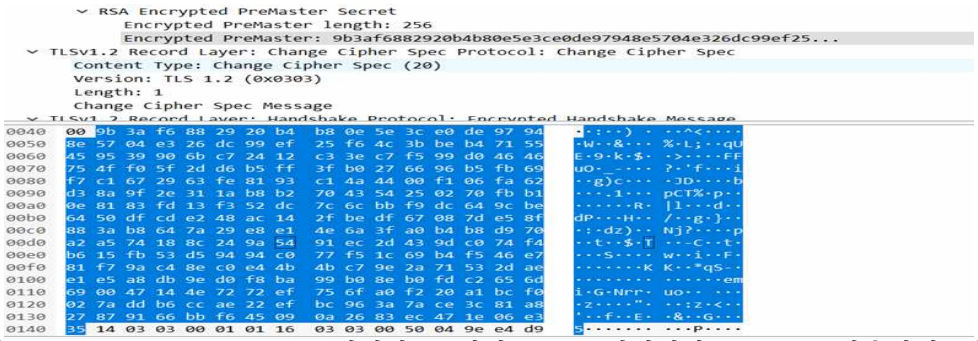
이 중에서 Client Key Exchange 패킷을 보내기 바로전 ClientHello와 ServerHello 패킷을 분석하여 통신에 필요한 키를 얻었습니다.



위의 사진은 통신의 일부분을 캡처한 것이며 TLS 통신을 하고, RSA를 통해 Premaster secret을 교환하며, 이를 이용하여 Master secret을 생성 후, Key block를 만듭니다. Key block에는 AES_128_CBC 이므로 서버와 클라이언트는 encrpytion key와 iv를 얻어 통신한다는 것을 확인하였습니다.

1. Premaster key 찾기

Client가 Server에게 보내는 ClientHello 패킷에서 encrpyted premaster key인 0x2ee25d7b1e01c33e95eaaef9a163a864c5cdd988a2dceb167b64223b5e48e9b735b228a35dacc5498798faa68430f503e4d6c44d2a181fa424371bf48b9e9ff4458a9ec429ce4827e43cdd4448faf0483ede1b2923d29cc0f56ad0f95914c303abd83c50c487c9b42f5587a5a61fa39b5558939f2eeef8d127b4cc352bb855fb9a7537876e5fa272ba912a77d5162fef640df1c1435950f676384853f5685fee591a80cb529f941b314d6256df1e77aac29c306f659bc4b86a729771b65b7e5d430c6567827bb42a2fff9542cc3000303f5230bdb21aa5272c304c343c5b1712ed40845165b020542e9f9e43c7c58b6c4242c1eef8d94732bf980e19bcaed 를 발견하였습니다.



암호화된 premaster secret은 문제에서 주어진 RSA 비밀키와 p, q를 사용하여 복호화하였습니다. 복호화한 premaster secret은

0x2ee25d7b1e01c33e95eaaef9a163a864c5cdd988a2dceb167b64223b5e48e9b735b228a35dacc5498798faa68430f503e4d6c44d2a181fa424371bf48b9e9ff4458a9ec429ce4827e43cdd4448faf0483ede1b2923d29cc0f56ad0f95914c303abd83c50c487c9b42f5587a5a61fa39b5558939f2eeef8d127b4cc352bb855fb9a7537876e5fa272ba912a77d5162fef640df1c1435950f676384853f5685fee591a80cb529f941b314d6256df1e77aac29c306f659bc4b86a729771b65b7e5d430c6567827bb42a2fff9542cc3000303f5230bdb21aa5272c304c343c5b1712ed40845165b020542e9f9e43c7c58b6c4242c1eef8d94732bf980e19bcaed 이며, PKCS#1 패딩을 사용하였으므로 원래의 premaster secret은 0x0303f5230bdb21aa5272c304c343c5b1712ed40845165b020542e9f9e43c7c58b6c4242c1eef8d94732bf980e19bcaed 입니다. 0x0303은 client version인 TLS 1.2임을 나타내며, 뒤의 수는 random한 수입니다.

premaster secret
0x0303f5230bdb21aa5272c304c343c5b1712ed40845165b020542e9f9e43c7c58b6c4242c1eef8d94732bf980e19bcaed

2. Session Hash 찾기

Session hash를 계산하기 위해서 도착한 순서대로 암호화된 handshake 이외의 모든 handshake 메시지를 첨부하여 계산합니다. 이 때의 handshake의 레코드 헤더는 포함하지 않고 메시지 및 메시지 헤더를 가지고 SHA256 해시함수를 사용하여 계산합니다. 아래의 값은 각 handshake hex stream 중 헤더 부분과 암호화된 handshake를 제외한 값입니다.

1.Client Hello

010000ab03035afa4a71973645bba7714b4ac91d3930c75191347bedbb3e98fdbe886111b3a1000026c02cc02bc030c02fc024c023c028c027c00ac009c014c013009d009c003d003c0035002f000a0100005c00500050100000000000a00080006001d00170018000b00020100000d0014001204010501020104030503020302020601060300230000001000e000c02683208687474702f312e3100170000001800060000a03020100ff01000100

2. Sever Hello, Certificate, Server Hello Done

200005103035afa49af70d1300d8825d9cf812fefd365ac52b083cf377c38d8c303989842bc200d2b000020ca7b1d58baecd2406e3f00a5ef94728603cfc8db509bcbc3da2788003c00000900170000ff010001000b00038c000389000386308203823082026a020900972dfc85011e71b0300d06092a864886f70d01010b0500308182310b3009060355040613024b52310e300c06035504080c0553656f756c310e300c06035504070c054e6f776f6e310d300b060355040a0c04536f6e673111300f060355040b0c08536f6e67536f6e67310e300c06035504030c05636869686f3121301f06092a864886f70d0109011612 6e616a616e613737406e617665722e636f6d301e170d313830353133313235393139 5a170d313930353133313235393139 5a308182310b30090603550406 13024b52310e300c06035504080c0553656f756c310e300c06035504070c054e6f776f6e310d300b060355040a0c04536f6e673111300f060355040b0c08536f6e67536f6e67310e300c06035504030c05636869686f3121301f06092a864886f70d01090116126e616a616e613737406e617665722e636f6d3082012230 0d06092a864886f70d01010105000382010f003082010a0282010100cc4efd665ab75ef6a69da2daaa50139c2772e7e11ff4f87889bc4e255b142348b936fc01329b15975da34caf6ec81fcbe7877c03bae7c21ec20298ad3b21d07a82db269c

696b7ae83c633e53aca5889bd81bca1d86e051a00a17248d861f8bf28b7671bc258ff95e078426adc8f1b2e729c787430a307ce0c0e73f050078cecc1e3a1e2c992efb4994f21ec277446a6dbb66bb5935f13763f4be5b73fd84416e1b1776b6d9ea2282eadc09af933415de96d82e925276f226cf9264f66742ef6e77e83a38d335955424d84acd696c37011dd65b5126664a5e1964301eb5b393d35e2ca3bfdaf361b97a1e3a748b4dd8db061f9078e526351a0588668d4bf308eb0203010001300d06092a864886f70d01010b050003820101000a3013c5880c28e511bbcddb27279a020c705bd20b58a093dc4fc92a0b329aabec972c2314b49248acecc64b561d1ee892c991afaaae2471e3e8a2033900489f0c367c17cde485697055d490cebed8fbbf88116a25047f8cec8a1b627041d5d8491d8a13f72eba74548d0536a4573981c104fb34cca14dd0303885bd333c6344c256278898d03bfe556e0b41beedb627130010313f6704ff6186e4d0da7d894bcd9724e36fc2c35e8ad9d436ec61c2bf51ee45cec7fdeefbf07cc65506b041951b43234e13ac3a8a64a7511fcf88654de02fe1fbe179cce550b111547c5ee1b7acb95e932bc73a1bdeb9c11d4098068e93a44ef818dbbfcd64a5ef870109b17a0e000000

4. Client Key exchange

1000010201009b3af6882920b4b80e5e3ce0de97948e5704e326dc99ef25f64c3bbeb47155459539906bc72412c33ec7f599d04646754ff05f2dd6b5ff3fb0276696b5fb69f7c1672963fe8193c14a4400f106fa62d38a9f2e311ab8b2704354250270fbb10e8183fd13f352dc7c6cbbf9dc649cbe6450dfcde248ac142fbedf67087de58f883ab8647a29e8e14e6a3fa0b4b8d970a2a574188c249a5491ec2d439dc074f4b615fb53d59494c077f51c69b4f546e781f79ac48ec0e44b4bc79e2a71532daee1e5a8db9ed0f8ba99b08eb0fdc2656d690047144e7272ef756fa0f220a1bcf0027addb6ccae22efbc963a7ace3c81a827879166bbf645090a2683ec471e06e335

>>모든 값을 넣고 SHA256을 계산하면

0xbcebc78de9c67da8ade8f46e0799afcf206285522af1e7ad89d3a17ab1cd04fe이 됩니다.

3. Master Secret 찾기

Premaster secret을 이용하여 Master secret을 구합니다. Extension : extended master secret이므로

master secret = PRF(premaster secret,"extended master secret", session hash)[0..47];

입니다.

PRF는 TLS 1.2 – IETF 문서를 참고하여 사용하였습니다.

P_hash(secret, seed) = HMAC_hash(secret, A(1) + seed) + HMAC_hash(secret, A(2) + seed) + HMAC_hash(secret, A(3) + seed) + ...

where + indicates concatenation. A() is defined as :

A() is defined as : A(0) = seed , A(i) = HMAC_hash(secret, A(i-1))

PRF(secret, label, seed) = P_<hash>(secret, label + seed)

HMAC_hash는 Cipher Suite를 통해 SHA 256임을 확인할 수 있습니다.

master secret = PRF(premaster secret,"extended master secret", session hash)[0..47];

= P_<hash>(premaster secret, "extended master secret" + session hash)

= HMAC_hash(premaster secret, A(1) + "extended master secret" + session hash)

+ HMAC_hash(premaster secret, A(2) + "extended master secret" + session hash)+...

(A(0) = 657874656e646564206d617374657220736563726574bcebc78de9c67da8ade8f46e0799afcf206285522af1e7ad89d3a17ab1cd04fe

A(1) =30422a9e9dc705fdff445bc85eeaf14f4b8391fb2bf268a838e13e13b8a845ab

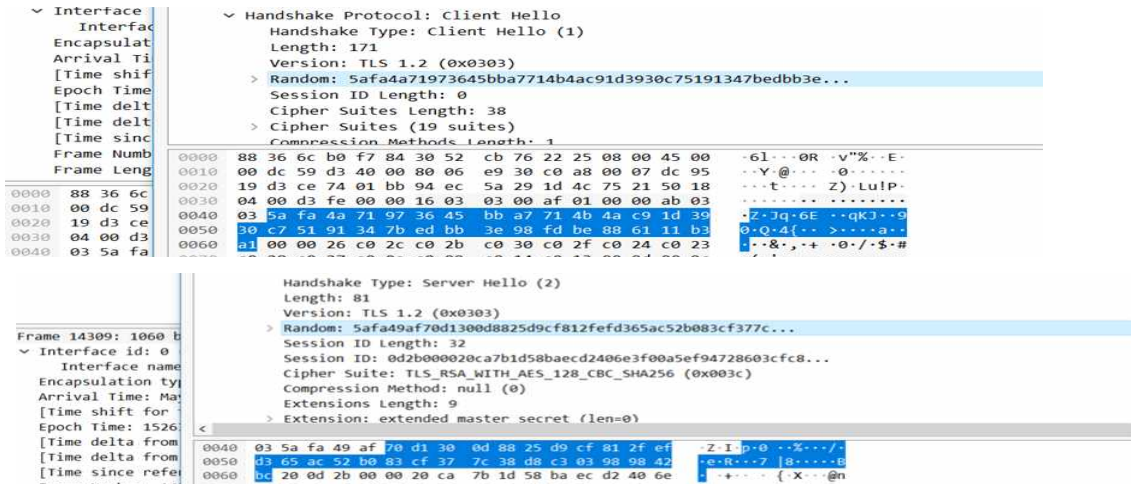
A(2) = 7649c4fece957118e8038ff5039600d87dd44dd640483db3e9dc2aa4d9dc02c8)

master secret =

0x6db26c792fcdef2290ba4892bf6ad5ab0e43ae319cb5c993be7775da3ceefd2858db22291cddf63ada8eaf1599de49d9

4. key block 찾기

먼저 server_random 값과 client_random 값을 패킷을 통해 찾습니다. 아래 사진은 두 값을 캡처한 사진입니다.



key block = PRF(master secret, " key expansion", server_random+client_random);

=PRF(0x6db26c792fcdef2290ba4892bf6ad5ab0e43ae319cb5c993be7775da3ceefd2858db22291cddf63ada8eaf1599de49d9, 0x6b657920657870616e73696f6e, 0x5afa49af70d1300d8825d9cf812fefd365ac52b083cf377c38d8c30398 9842bc + 0x5afa4a71973645bba7714b4ac91d3930c75191347bedbb3e98fdbe886111b3a1)

(A(0) = seed = 0x6b657920657870616e73696f6e5afa49af70d1300d8825d9cf812fefd365ac52b083cf377c38d8c3
03989842bc5afa4a71973645bba7714b4ac91d3930c75191347bedbb3e98fdbe886111b3a1

A(1) = 0xce152d9ff9a74e446ffeb67a681dade9b0e4868dceda29a2ec682aa97e57aecf

A(2) = 0xeada02a1e803af399e18428314d0bd9b66792a751958d8f52744053df045af6c

A(3) = 0xa2044dfa444771083268d8322774a394a4455386fd768e23c9ca50e295df3d86

A(4) = 0x2fad6c28243d65c7e51a8197d325d798b92474acf51a1cffd82b803d55dcb53f)

Key block = 0x3b82b13d1a0a5f51094005ca0dc2586d98b69ef4589b5109229fcdc8b161de05(client_MAC_key)
+ 0xf3aa5489548758c49ab9d851d5594e638c6eafeb07a728ca1b212b9d53e26475(server_MAC_key)
+ 0x647592711ae5e26f140c45d80323e922(client_enc_key)
+ 0xd6b2eb475c92649e9c977475eec403a7(server_enc_key)
+ 0x7356e0948a0544d5d8bfc01441a4f3d6(client_IV)
+ 0x82a5f1a84453659c7c9df430eec139a1(server_IV)

5. 암호문 C를 복호화 하여 P를 구하고, Character Array로 출력한 문장을 구하라.

맨 처음에서 알 수 있듯이 패킷의 암호화는 AES128 CBC운영모드이고, Server가 Client에게 보내는 암호문이므로 Server의 키와 IV 값을 통해 복호화 하면 평문의 값을 알 수 있습니다.

F71C91A475189AFDF7BAF937354E84FF85D7B81CCE25DDA286C5766C70972AFE3E51EDCEDF566D5E8F07A8E6557FB87A4EC2B0A151A5EAE7A95082B7251E8AEB

>> Congratulations! You solved this problem.