**HOMEWORK**

2011140138

황태림

1. **Describe BEAST attack scenario in detail, which is the chosen-plaintext attack on CBC mode in TLS 1.0 (or prior SSL versions), to decrypt a message byte-by-byte**

BEAST는 Browser Exploit Against SSL/TLS 의 약어이다. Chosen plaintext attack 기법이다. CBC를 사용하는 SSL/TLS에서 CBC의 첫 plaintext block을 encrypt할 때 필요한 Initialization Vector(IV)를 target으로 한다. Attack scenario를 다음과 같이 가정해본다.

**TLS 1.0**

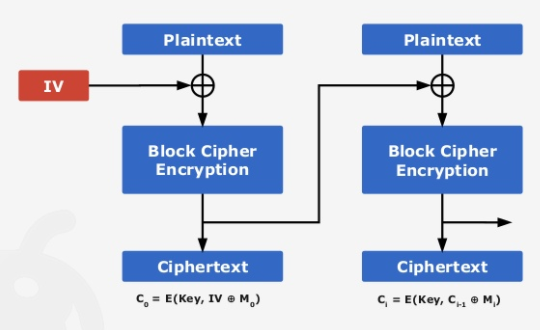
**CBC**

**Alice**

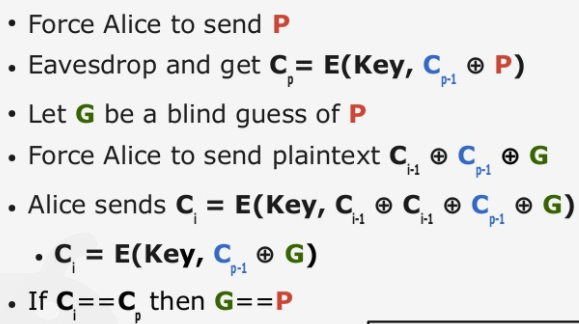
**Bob**

**Eve**

Alice와 Bob이 TLS 1.0(CBC mode)로 메시지를 주고받는데 이 때, Eve 공격자가 공격을 시도한다. Eve는 Plaintext[16bit]를 알고 싶고, 이를 위해 Alice와 Bob 간의 주고받은 encrypted된 data를 엿듣는다. 또한 Eve는 Alice가 선택된 특정 plaintext를 보낼 수 있도록 강제할 수 있다. CBC는 IV의 비밀 유지가 매우 중요한데, 만약 IV가 공격자 Eve에게 알려진다면 추후에 encrypt 되는 모든 plaintext block 또한 변조가 가능하고, 원래의 plaintext를 알 수 있다.

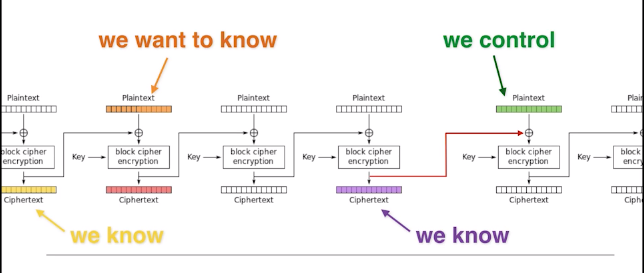


위의 그림과 같이 CBC는 구성이 되어 있다. 이 때, 공격자 Eve는 Alice가 특정 plaintext P 를 보내도록 한다.

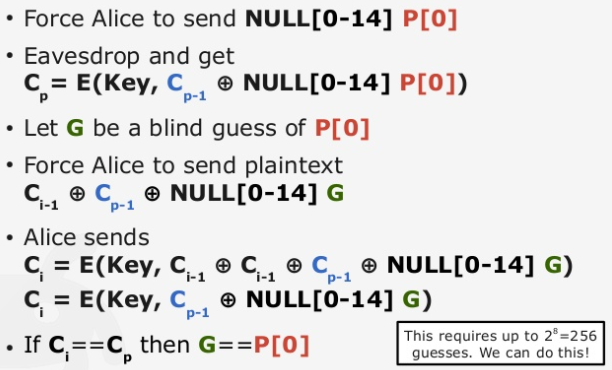


CBC chosen plaintext attack에 대해 조사한 결과 위와 같은 자료를 찾을 수 있었다. Alice에게 공격자 Eve는 chosen plaintext인 P를 보내고, Encrypt된 Cp를 얻게 된다. 또한, Cp-1의 값도 알고 있는 상태이다. 이제, 다른 위치의 Ci-1과 알고자 하는 plaintext G가 있다고 하자. 이 때, P를 Ci-1 XOR Cp-1 XOR G라 하고, Ci를 구하면 Ci = E(Key, Cp-1 XOR G)로 정리가 된다. G의 값을 수정하면서 Ci == Cp가 될 경우, 그 때의 G가 우리가 알기 원하던 plaintext G이다.

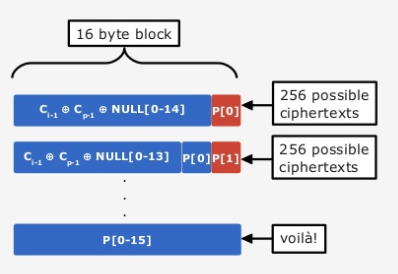
조금 더 자세한 그림으로 본다면 다음과 같다. 알고자 하는 plaintext block의 이전 ciphertext block을 안다면, 혹은 IV를 안다면, XOR 연산의 특징으로 IV XOR IV = cancel 로 인해 원하는 bit를 추측 가능하다.



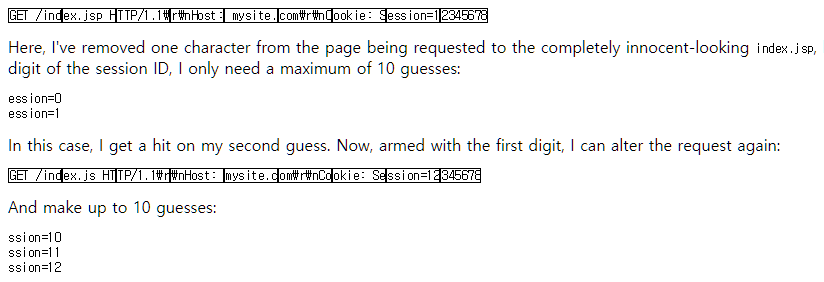
다음은 byte-by-byte BEAST attack에 관한 내용이다.



16byte의 plaintext block이 있다면, 1byte의 특정 byte와 나머지 15byte의 NULL 또는 반복되는 문자나 알려진 문자를 앞에 끼워 넣은 후, 위의 CBC attack scenario대로 진행하면, 원하는 plaintext block의 맨 앞자리의 값을 알게 된다.



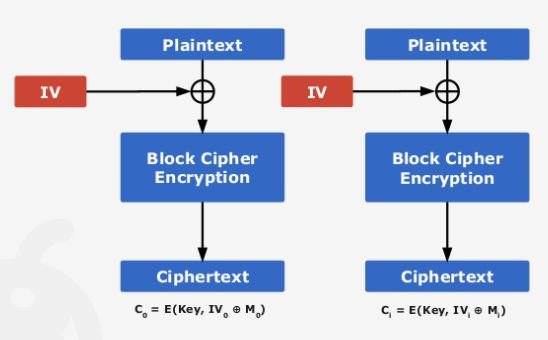
이렇게 byte-by-byte로 하나씩 추측하며 확인하면 원하는 전체 plaintext block을 알게 된다.



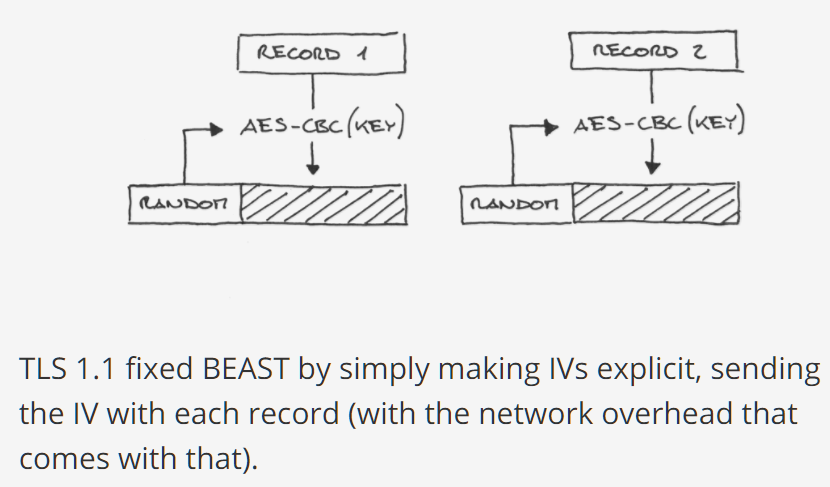
패킷을 이용한 예를 들면, index.html로 보내던 주소를 index.jsp로, 하나의 글자를 줄이면(-1 byte) Session 뒤의 키를 알기 위해 ession=?로 적은 경우의 수로 추측해 나갈 수 있다. 1byte = 2^8 경우의 수를 가지므로 1byte를 알기 위해 최대 256번의 시도를 하게 된다.

1. **Find the countermeasure fixed in TLS 1.1, and demonstrate why this is the case (how it can solve the vulnerability)**

TLS 1.1부터는 위와 같은 BEAST attack으로부터 문제가 없도록 다른 방식의 encrypt mode를 사용한다. IV를 처음 plaintext block에만 적용하고 이후에는 이전의 ciphertext block을 XOR하는 TLS 1.0과는 달리, TLS 1.0은 매 plaintext block 의 encryption에 strong random string R을 IV로 사용한다. 이를 그림으로 표현하면 다음과 같다.



이를 통해 기존의 XOR 연산으로 인하여 발생하는 문제(Ci-1 XOR Cp-1 XOR G = P 에서 Ci를 구할 때의 Ci-1 XOR Ci-1 = cancel)를 없앨 수 있다. 첫 IV는 mask라 부른다.



다음과 같이 PRNG을 통해 fixed block F를 생성 후 plaintext에 붙여 보내는 방법도 존재한다. TLS 1.1은 3가지의 알고리즘(random R로 IV 대신 넣은 후 encrypt, random R을 plaintext에 붙인 후 encrypt[fixed mask 사용, 이전의 record를 mask로 사용])중 어느 방법을 사용하더라도 decryption은 같다.

1. **출처**

<https://www.youtube.com/watch?v=-_8-2pDFvmg> - BEAST: An Explanation of the CBC Attack on TLS, David Wong

<http://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art027> - An Illustrated Guide to the BEAST Attack, Joshua Davies

<https://tools.ietf.org/html/rfc4346> - The Transport Layer Security (TLS) Protocol Version 1.1, T. Dierks, April 2006

<https://www.slideshare.net/danrlde/20120418-luedtke-ssltlscbcbeast> - BEAST attack on SSL/TLS explained, Dan Lüdtke, Mar 1, 2015

<https://tools.ietf.org/html/rfc2405> - The ESP DES-CBC Cipher Algorithm Explicit IV, C.Madson, November 1998

<https://blog.cloudflare.com/tls-nonce-nse> - TLS nonce-nse, Filippo Valsorda, 12 Oct, 2016