인터넷에서 사용되는 여러 기술 : HTTPS 7

조인준
KBS 미디어기술연구소 차장

지난 편까지 TCP 기반 통신에서 암호화된 보안 통신을 위한 SSL/TSL 기술에 대해 설명드렸습니다. SSL/TLS에 대해 다시 간략한 용어 설명을 설명드리면. SSL은 Secure Socket Layer의 약자로 1994년 넷스케이프(Netscape)에 의해 개발된 인터넷 보안 기술입니다. 넷스케이프가 1999년 국제 인터넷 표준화 기구 IETF(Internet Engineering Task Force)로 SSL의 관리를 이전하면서 TLS(Transport Layer Security)로 명칭이 바뀌었으나 예전 명칭인 SSL이 계속 사용되면서 TLS만 사용하거나 SSL/TLS로 병기하기도 합니다. 이후 SSL/TLS는 [그림 1]과 같이 몇몇 버전을 거치며 발전해왔습니다.

그림 1. SSL/TSL 버전 및 연도

SSL/TLS 기술의 가장 최신 버전은 2018년에 만들어진 TLS 1.3입니다. TLS 1.3은 이전 버전인 TLS 1.2에서의 보안 결함과 성능 이슈를 개선하기 위해 설계되었고 이에 따라 TLS 1.2 대비 여러 변화와 개선이 있습니다. 이런 변화와 개선에서 가장 중요하게 작용한 원칙은 보안성과 단순화이며 이를 위해 하위호환(Backward Compatibility)을 포기한 부분도 있습니다. 많은 부분에서 TLS 1.3은 이전 버전보다 더 보안성이 높으며 효율적인 프로토콜이기 때문에 많은 웹 서비스에서 채택되고 있습니다. 그러나 아직 TLS 1.3이 도입되지 않은 클라이언트나 서버는 TLS 1.3을 지원하지 않을 수 있습니다. 이 경우 이전 버전인 TLS 1.2 등을 사용하여 보안 통신을 제공할 수 있습니다.

TLS 1.3의 변경사항 중 중요한 내용을 요약하면 다음과 같고 이러한 변화들이 TLS 1.3을 이전 버전보다 더 강력하고 효율적인 보안 프로토콜로 작동하도록 만들었습니다.

• 핸드셰이크(Handshake) 단순화

TLS 1.3은 핸드셰이크 프로토콜을 단순화하여 불필요한 핸드셰이크 단계를 제거했습니다. 이로써 핸드셰이크의 지연 시간을 줄이고 연결 설정 시간을 단축했습니다.

• Cipher Suite 단순화

TLS 1.3은 이전 버전의 다양한 Cipher Suite(암호 집합)를 단순화하여 더 간결하고 보안성이 높은 암호화 알고리즘 세트를 제공합니다.

• **Forward Secrecy**

이전 버전과는 달리 TLS 1.3에서는 기본적으로 Forward Secrecy(순방향 비밀성)가 제공됩니다. 이는 세션 키가 단기적으로만 사용된 후 파기되어 공격자가 키를 획득하더라도 이전에 수행된 통신을 해독할 수 없습니다. 이는 사용자의 데이터를 보호하고, 보안을 유지하는 데 큰 역할을 합니다.

• **0-RTT 모드**

TLS 1.3에서는 0-RTT(Zero Round Trip Time) 모드를 도입하여 이전에 수행된 핸드셰이크에서 얻은 정보를 재사용하여 최초 연결을 설정하는 데 필요한 라운드 트립을 제거합니다. 이는 초기 연결 설정 속도를 높이는 데 도움이 됩니다. 하지만 0-RTT 모드는 이전에 사용한 세션 정보를 재사용하기 때문에 이를 악용한 보안 공격 등의 위협이 발생할 수 있습니다. 따라서 0-RTT 모드를 사용하는 경우에는 추가적인 보안 조치가 필요합니다.

• **암호화의 강화**

TLS 1.3은 이전 버전에서 사용되던 일부 취약한 암호화 알고리즘을 제거하고 보안성이 높은 알고리즘으로 대체하여 보안을 강화했습니다.

이와 같은 변화 중 가장 주목해야 할 부분은 핸드셰이크 및 Cipher Suite의 단순화입니다. 핸드셰이크 단순화를 통해 속도가 크게 향상되고 Cipher Suite 단순화를 통해 오래된 암호화 방식을 폐기함으로써 보안성을 강화했기 때문입니다. 이 두 가지 가장 중요한 변화 중, 우선 핸드셰이크 단순화를 살펴보겠습니다.

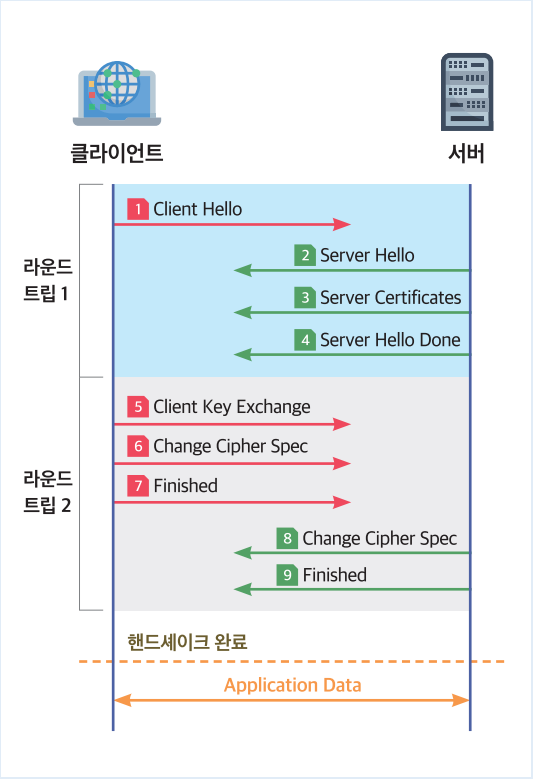

그림 2. TLS 1.2 핸드셰이크

[그림 2]는 TLS 1.2 핸드셰이크 과정을 보여주고 있습니다. [그림 2]의 TLS 1.2 핸드셰이크는 ① Client Hello ~ ④ Server Hello Done까지의 첫 번째 라운드 트립(Round Trip)과 ⑤ Client Key Exchange ~ ⑨ Finished까지의 두 번째 라운드 트립으로 이루어진 2-RTT(Round Trip Time) 구조를 가지고 있습니다. 본 연재의 앞부분에서 다루어져 [그림 2]의 내용이 생소한 독자분들을 위해 [그림 2]를 간단히 요약·설명하면 다음과 같습니다.

클라이언트는 서버에 ① Client Hello 메시지를 전송하여 핸드셰이킹을 시작하며 이 메시지 안에는 클라이언트가 지원하는 SSL/TSL의 최상위 버전 정보, Cipher Suites(클라이언트가 지원하는 암호화 방식들의 리스트), Session ID 등의 정보가 포함되어 있습니다. 서버는 Client Hello에 응답으로 ② Server Hello를 보내며 Client Hello와 동일한 항목을 포함하고 있습니다. 그리고 이어서 ③ Server Certificate 메시지를 통해 클라이언트에게 자신의 인증서를 보내며 서버 인증서와 중간 인증서 등이 포함되어 있습니다. 루트 인증서도 추가로 보낼 수 있지만, 보통 클라이언트가 이미 신뢰하는 루트 인증기관의 공개키를 알고 있기 때문에 이는 필요에 따라 보내게 됩니다. Server Certificate 메시지 이후에는 서버

가 지금까지 필요한 메시지를 모두 보냈음을 나타내기 위한 ④ Server Hello Done 메시지를 보냅니다. Server Hello Done 메시지를 받으면 클라이언트는 Pre-master Secret이라는 것을 생성한 후 서버로부터 받은 인증서에 포함된 공용키로 암호화하여 ⑤ Client Key Exchange 메시지를 통해 보냄으로써 이후의 통신에서 사용될 대칭키를 생성하는 데 이용함과 동시에 서버가 인증서의 실제 소유주가 맞는지 확인하게 됩니다. 서버가 인증서에 포함된 공용키에 대응되는 사설키를 소유하고 있다면 클라이언트가 암호화하여 보낸 Pre-master Secret을 정상적으로 복원하여 이후의 절차에 이상 없이 대응할 수 있기 때문입니다. 클라이언트는 Client Key Exchange 메시지를 보낸 후 핸드셰이크 마무리를 위해 지금까지 보낸 보안 관련 데이터들을 대칭키로 암호화한 정보를 담은 ⑦ Finished 메시지를 서버에 보냅니다.

클라이언트의 Finished 메시지를 받은 서버는 ⑧ Change Cipher Spec 메시지를 보내서 서버가 성공적으로 대칭키를 생성했으며 이후의 메시지는 지금까지의 과정을 통해서 정해진 암호화 방식으로 진행할 것임을 알립니다. 또한 서버도 클라이언트와 마찬가지로 지금까지 보낸 보안관련 데이터들을 대칭키로 암호화한 정보를 담은 ⑨ Finished 메시지를 보내서 클라이언트와 서버가 같은 데이터에 기반하여 암호화된 통신을 수립했음을 서로 확인합니다. 이런 과정을 통해 핸드셰이크가 완료되면 서버와 클라이언트 간 암호화된 통신이 본격적으로 시작됩니다.

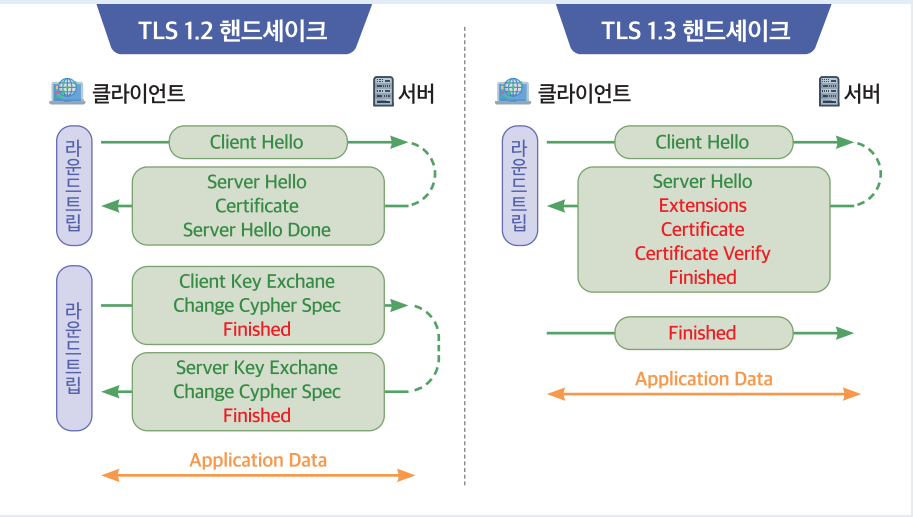

그림 3. TLS 1.2 vs TLS 1.3 핸드셰이크 비교

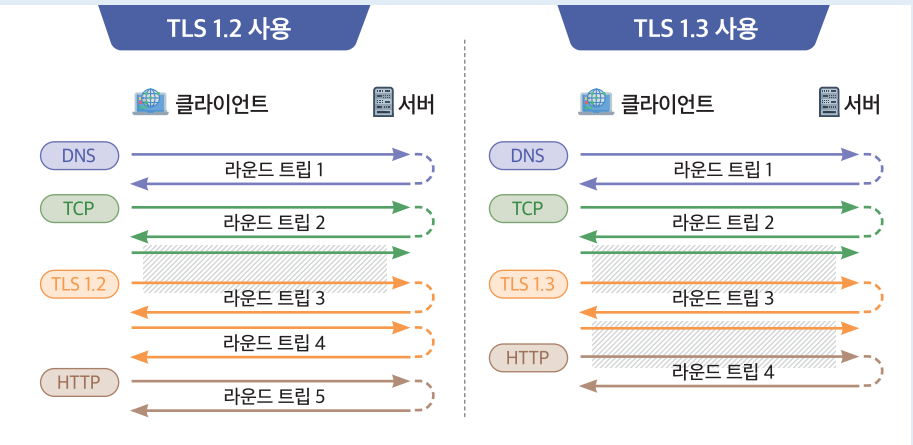

그림 4. 웹 사이트 접속부터 첫 데이터 수신까지 라운드 트립 비교

TLS 1.2의 2-RTT 구조의 핸드셰이크가 TLS 1.3에서는 1-RTT 구조로 변경되었으며 [그림 3]에 이를 비교하였습니다. [그림 3]은 라운드 트립을 중심으로 메시지 교환을 정리했으며 주고받는 메시지 중에 암호화된 메시지는 붉은색으로 표시하였습니다. 라운드 트립으로 정리되었기 때문에 [그림 2]에서 같은 방향으로 진행되는 연속된 메시지는 [그림 3]에서 하나의 화살표로 통합되어 표시되어 있습니다.

[그림 3]에서 알 수 있듯이 TLS 1.3에서는 TLS 1.2 대비 라운드 트립이 1회로 줄었으며 메시지의 더 많은 요소가 암호화된 상태로 핸드셰이크가 진행됩니다. TLS 1.3 핸드셰이크의 라운드 트립 단순화가 TLS 1.2 대비 전체적 관점에서 어느 정도의 이점이 생기는지 TLS를 사용하는 HTTPS로 웹 사이트에 접속할 경우 시작부터 최초의 웹 사이트 관련 데이터를 받기까지의 과정을 [그림 4]에 비교하였습니다.

[그림 4]는 인터넷상의 서버에 접속하여 요청한 데이터를 받기까지의 과정을 라운드 트립 중심으로 정리한 그림입니다. 라운드 트립 관점에서 정리된 그림이므로 화살표 하나가 한 번의 메시지 전송이 아닌 동일한 방향으로의 복수 메시지 전송일 수 있는 점을 유의하고 보시기 바랍니다.

인터넷상의 서버에 접속하기 위해서는 tech.kobeta.com(방송과 기술 홈페이지 주소) 같은 문자로 된 주소가 아닌 IP 주소가 필요합니다, 이를 위해 DNS(Domain Name System) 서버에 접속하여 tech.kobeta.com이 가리키는 IP 주소를 얻어 옵니다. 이 과정에 한 번의 라운드 트립이 필요합니다. 그다음은 HTTP의 통신 방식인 TCP 통신을 위한 TCP 핸드셰이크 단계를 거쳐야 하며 이때도 한 번의 라운드 트립이 필요합니다. [그림 4]에서 TCP 핸드셰이크 단계에서 한 번의 라운드 트립 이후 클라이언트가 다시 한번 메시지를 보내고 이어서 TLS 핸드 셰이크를 시작하는데 이 경우 앞서 설명되었듯이 라운드 트립은 화살표 하나가 한 번의 메시지 전송이 아닌 동일한 방향으로의 복수 메시지 전송일 수도 있으므로 빗살무늬로 TCP 핸드셰이크의 마지막 메시지와 TLS 핸드셰이크의 시작을 같이 묶어 연계된 동작으로 봐도 무리가 없습니다.

이렇듯 이후 단계의 라운드 트립과 연계될 수 있는 클라이언트 → 서버로의 메시지 전송은 [그림 4]에서 빗살무늬로 표시하였습니다. [그림 3]과 같이 TLS 1.2 핸드셰이크는 2번의 라운드 트립이 필요하고 TLS 1.3은 한 번의 라운드 트립만으로 핸드셰이크가 완료됩니다. 이후 클라이언트가 HTTP 요청을 서버에 보내고 서버로부터 첫 데이터를 받는 라운드 트립까지 더하여 서버와의 실제 데이터 통신 개시까지의 라운드 트립 횟수를 셀 수 있습니다. TLS 1.2는 최초 데이터 수신까지 5번의 라운드 트립이 필요하고 TLS 1.3은 4번의 라운트 트립이 소요됩니다.
라운드 트립 횟수로 보면 1번의 차이뿐이지만 %로 환산하면 5회에서 4회로 최초 웹 사이트 데이터 수신까지의 라운드 트립 횟수가 TLS 1.3에서 20% 감소함을 알 수 있습니다. 이는 연결 설정에 필요한 시간이 감소함과 동시에 수천~수만 건을 처리하는 서버 입장에서도 이점이 있습니다. 더불어 이전의 TLS 버전에서는 보안 상태를 설정하기 위해 2번의 라운드 트립이 필요했던 것에 비해 TLS 1.3은 한 번의 라운드 트립만 필요하므로 공격자가 연결을 간섭하거나 공격하기가 어려워졌습니다. 이는 [그림 3]의 강화된 핸드셰이크 관련 데이터의 암호화와 더불어 TLS 1.3의 보안성을 더욱 강화하는데 기여합니다.

TLS 1.3에서는 1-RTT 핸드셰이크 외에도 0-RTT 핸드셰이크 방식을 지원합니다. 이는 서버와 클라이언트 사이에 이미 공유된 정보를 사용하여 새로운 연결을 설정하는 방식인데 예를 들면 클라이언트가 이전에 방문한 적이 있는 웹 사이트에 다시 접속할 때 서버가 클라이언트에게 당시 전송했던 인증서와 같은 보안 정보를 다시 보내지 않고 새로운 연결을 수립하는 식으로 가능해집니다. 하지만 세상에 공짜는 없듯이 0-RTT를 통해 성능을 향상하고 연결 설정 시간을 줄일 수 있지만 이전에 공유된 데이터를 통한 연결 수립에 의해 보안 측면에서는 주의를 기울여야 하며 추가적인 검증 및 보안 조치가 필요합니다.

지금까지 TLS 1.3의 가장 큰 변화 중 하나인 핸드셰이크 관련 내용을 살펴보았습니다. 다음 편에서는 다른 하나의 큰 변화인 Cipher Suite에 관한 내용을 다루겠습니다.

P.S.

C군이 여러분께 전하는 내용 중 전문적 성격이 짙은 것은 엄밀한 언어를 사용하여 설명하기에는 한계가 있습니다.
본 내용은 설명하는 대상에 대한 전체적 맥락의 이해에만 이용하시고, 그 이상은 권위 있는 전문자료를 참고하시기 바랍니다.