인터넷에서 사용되는 여러 기술 : HTTPS 2

조인준
KBS 미디어기술연구소 차장

HTTP(Hypertext Transfer Protocol)의 보안강화 버전인 HTTPS(Hypertext Transfer Protocol Secure)에 대한 두 번째 이야기입니다. 지난 편에서는 HTTPS의 보안 기술인 SSL/TLS를 설명하기 위한 첫 단계로 대칭키와 비대칭키에 관한 기초적인 내용을 소개하며 비대칭키는 연산속도는 느리지만 보안성이 강하고, 대칭키는 연산속도는 빠르지만, 보안성이 약한 특성이 있음을 알려드렸습니다. 암호화(Encryption)를 통한 실용적인 보안 기술을 완성하기 위해서는 속도와 보안성 모두 중요하기에 비대칭키로 데이터양이 적은 대칭키를 암호화하여 안전하게 공유하고 이후의 데이터양이 큰 메시지는 대칭키로 암호화하여 연산속도와 보안성을 양립한 암호화 방식이 사용되고 있다는 것까지 말씀드렸습니다.

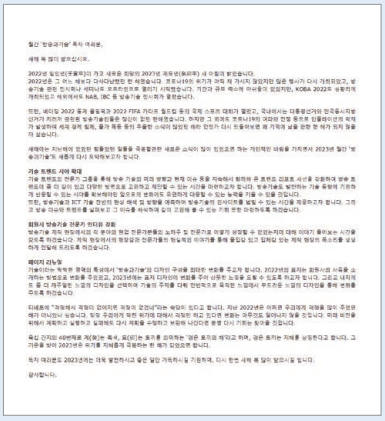

그림 1. 방송과 기술 홈페이지 편집장 인사말 본문

암호화와 함께 서명(Signature)도 보안에서 중요한 요소이고 비대칭키와 대칭키를 병용하는 암호화 방식과 마찬가지로 서명에도 속도와 보안성을 양립하기 위한 방식이 존재합니다. 이 방식을 이해하기 위해서는 우선 '해시(Hash) 알고리즘'에 대해 알아야 합니다. 해시 알고리즘은 임의의 크기의 데이터를 고정된 크기의 고유한 값으로 매핑하는 알고리즘이며, 두 해시값이 다르다면 그 해시값에 대한 원래 데이터도 다릅니다. 하지만 그 역은 성립하지 않아서 서로 다른 데이터가 같은 해시값을 가질 수는 있습니다. 그래서 해시값으로 원래의 데이터를 복원하는 것은 불가능합니다. 해시 알고리즘에는 MD5, SHA-1, SHA-256, SHA-3 등의 방식이 있습니다.

해시 알고리즘에 자체에 대한 설명보다는 해시란 무엇이가에 관한 결론만 간단히 말씀드리겠습니다. 해시 알고리즘으로 '방송과 기술'을 해시하면 MD5 방식으로는 32자의 '61ad99a9e50ea0ae938819c1abcd639e'이고 SHA-256 방식으로는 64자의 'dc22ba537235868b2e8dbb989c6e793984aae939e265654db300bfbddf83e0e5'이 됩니다. 공란까지 포함해서 고작 여섯 자를 해시 했는데 32자나 64자로 늘어나니 뭔가 불합리한 것 같은 생각이 듭니다. 하지만 다른 예를 하나 보시면 생각이 달라질 겁니다.

[그림 1]은 방송과 기술 홈페이지의 편집장 인사말입니다. 내용이 중요한 것이 아니고 전체 텍스트가 어느 정도의 양인지가 중요하므로 그림은 분량을 알아볼 수 있을 정도로의 크기로만 첨부했습니다. [그림 1]의 방송과 기술 사이트의 편집장 인사말 전체를 있는 그대로 해시하면 MD5 방식으로는 32자의 '352f696815190ffb462a9795c3c7b1fa'가 되고 SHA-256 방식으로는 64자의 '09cc67fe11c08a183ad4513c1f76ac34c074d69a2e7c4c7138e9f5e1cdce3635'가 됩니다. 앞서 설명해 드린

대로 공란 포함 여섯 자의 '방송과 기술'과 수 백자가 넘는 편집장 인사말의 해시값은 길이가 같습니다. 그렇다면 이런 해시를 어떻게 이용하여 속도와 보안성을 양립한 서명기술을 구현할까요? 서명의 기술적 원리의 환기를 위해 이전 편에서 설명해 드린 내용을 다시 한번 요약하면 다음과 같습니다.

서명은 다른 사람에게 메시지를 보낼 때 이것이 자신이 보낸 것이고 중간에 누군가로 인해 변조된 것이 아님을 보장하기 위한 것으로, 메시지를 자신의 사설키로 암호화하여 상대에게 보내고 이 암호화된 메시지를 받은 상대는 보낸 이의 공용키를 이용해서 암호를 복원합니다. 사설키로 암호화된 메시지는 그 사설키에 대응되는 공용키로밖에 해독이 불가하므로, 공용키로 복원된 메시지가 정상적인 메시지라면 이는 그 공용키에 대응되는 사설키를 가진 사람이 보낸 메시지가 맞습니다. 또한, 암호화된 메시지 전달과정 중간에 이 메시지를 암호화한 사설키에 대한 정보가 없는 누군가가 암호화된 상태의 메시지를 변조했다면 공용키를 이용해 해독한 내용은 말이 안 되는 메시지가 될 수밖에 없습니다. 이렇게 보낸 사람의 신원인증(Authentication) 및 메시지가 변조되지 않았다는 무결성(Integrity)을 보장하기 위한 것이 서명입니다.

하지만 긴 메시지를 주고받을 때 사설키와 공용키를 사용하는 비대칭키 방식으로 전체 메시지를 암호화하여 서명할 경우 계산량 증가에 따른 속도 저하의 부담이 있을 수 있습니다. 이런 부담을 줄이려면 이전 편에서 설명해 드린 비대칭키와 대칭키를 혼용한 암호화처럼 보안성과 처리속도를 양립시키는 기술이 서명에도 필요합니다. 보안성을 높이려면 비대칭키를 사용하여야 하는데 비대칭키는 데이터의 길이가 길어질수록 계산 속도에서 취약함을 노출하기 시작하니, 비대칭키를 사용하되 암호화의 대상이 되는 데이터 길이를 줄여서 서명에 이용하는 방법을 고안하면 문제가 해결될 수도 있을 것입니다. 이런 아이디어에 기반해서 만들어진 서명 방식이 해시 알고리즘을 이용하여 데이터를 줄이고, 여기에 비대칭키를 적용하여 서명의 보안성을 높이는 [그림 2]~[그림 6]의 방식입니다.

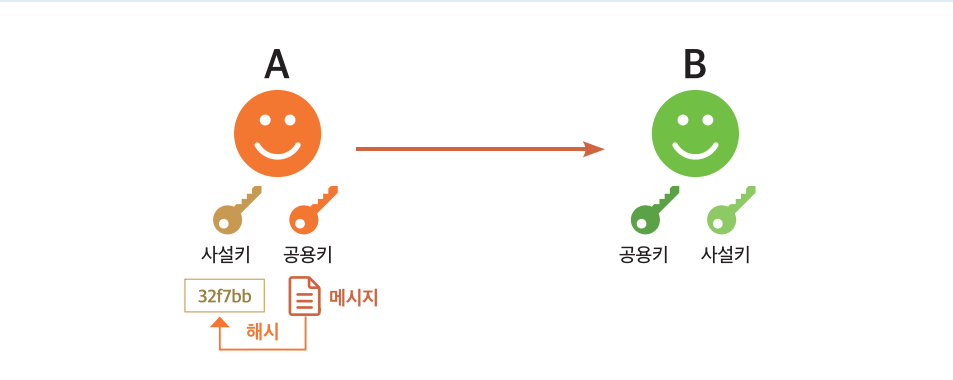

그림 2. 메시지와 해시값

[그림 2]와 같이 보내려는 메시지를 해시하면 [그림 1]을 이용해 보여드린 예와 같이 메시지 본문보다 훨씬 짧은 일정한 길이의 문자열로 바꿀 수 있습니다. 메시지에 비해 비약적으로 짧아진 해시값은 원래 메시지를 수정한 후 다시 계산할 경우 메시지 변경 전의 해시값과 다른 값을 가집니다. 이로부터 해시값을 메시지의 지문처럼 사용하는 것이 가능합니다. 만약 [그림 3]처럼 A가 긴 메시지를 해시하여 그 값을 자신의 사설키로 암호화한다면 어떨까요? 누군가 몰래 메시지를 변경하면 이의 해시값도 같이 바꾸어야 완벽히 속일 수 있습니다. 다시 말해 누군가 메시지를 조작하면 조작된

메시지의 해시값을 A가 가진 사설키로 암호화하여야 이를 A의 공용키로 해독했을 때 메시지의 해시값과 일치하여 완벽하게 속일 수 있습니다. 그렇지만 A의 사설키는 A가 유일하게 가지고 있는 키이고, 이것이 노출될 가능성이 없다면 암호화된 해시값을 조작된 메시지와 일치하게 바꿀 방법은 현실적으로 없습니다. 그러므로 [그림 3]의 메시지의 해시값을 사설키로 암호화한 것이 사실상의 서명이 되고, 이는 A의 공용키를 가진 누구나 해독하여 확인할 수 있습니다. 이렇게 메시지 전체를 사설키로 암호화하지 않고 해시값을 암호화하는 방식으로 서명의 계산 부담을 줄이고 속도를 높일 수 있습니다.

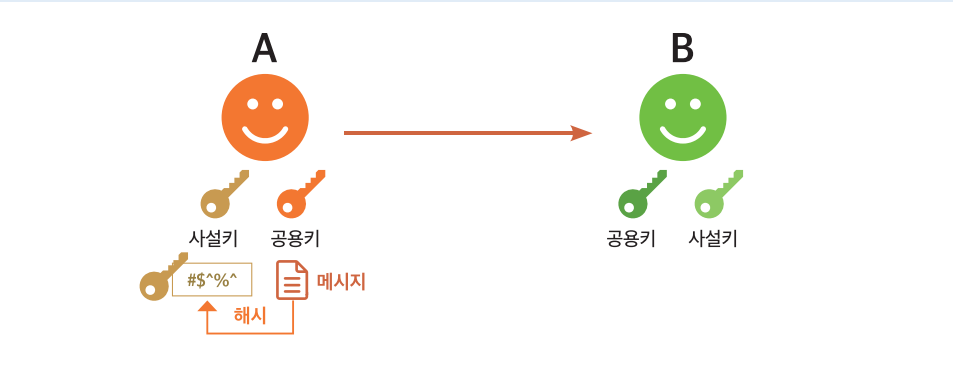

그림 3. 사설키를 이용한 해시 값의 암호화(서명)

[그림 3]과 같이 메시지를 해시한 후 이 해시값을 사설키로 암호화(A의 서명에 해당)한 것을 메시지와 함께 [그림 4]와 같이 B에게 보내면 B는 [그림 5]와 같이 A의 공용키를 이용하여 암호화된 해시값을 복원합니다.

그림 4. 메시지와 사설키로 암호화된 해시 값 전송

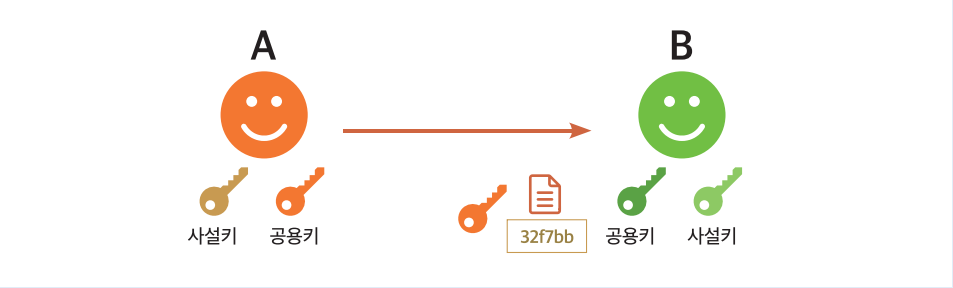

그림 5. 공용키를 이용한 해시 값 복원

A가 보낸 메시지와 A의 사설키로 암호화된 해시값을 A의 공용키를 이용하여 복원한 B는 마지막 확인으로 [그림 6]과 같이 스스로 메시지를 해시하여 그 값을 A가 보낸 해시값과 비교합니다.

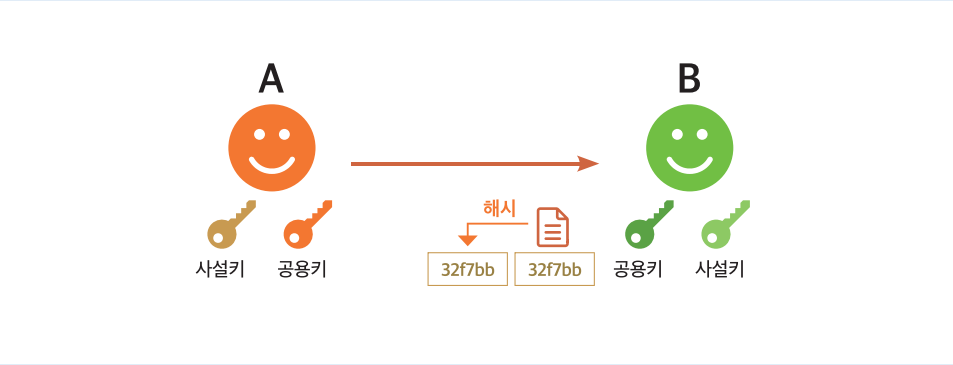

그림 6. 메시지에서 계산된 해시값과 해독된 해시값 비교

B가 직접 메시지를 해시한 값과 A의 공용키로 복원한 해시값이 같다는 것은 다음 두 가지를 보장합니다.

☑ 중간에 누군가 몰래 메시지를 변경하지 않았다

A가 자신의 사설키로 암호화하여 보낸 해시값은 A의 공용키로 해독이 가능합니다. 만약 누군가 메시지를 변경하고 이 사실을 숨기려 같이 보내는 암호화된 해시값도 변경하려면 A의 사설키가 필요합니다. 하지만 A의 사설키는 A만이 가지고 있습니다. A가 가진 사설키가 없다는 것은 A가 어떻게 해시값을 암호화한지 모르는 것과 같습니다. 이는 사실상 변경한 메시지의 암호화된 해시값을 만들 방법이 없다는 것과 같습니다. 그러므로 A의 공용키로 해독한 수신 메시지의 해시값과 B가 메시지로부터 직접 계산한 해시값이 같다면 누군가 중간에 이 메시지를 변경하지 않았다는 것이 되며 인터넷 보안에서의 무결성(Integrity)에 해당합니다.

☑ B가 받은 메시지는 A가 보낸 것이 맞다

A의 공용키로 복원한 해시값이 메시지로부터 직접 계산한 해시값과 일치하면 해당 메시지는 A가 보낸 것이 맞습니다. A의 사설키로 암호화한 것은 A의 공용키로밖에 복원이 불가능한데 역으로 A의 공용키로 제대로 복원이 되었다면 이는 A의 사설키로밖에 암호화가 안 되고, A의 사설키는 말 그대로 세상에서 A만 가지고 있기 때문입니다. 이는 인터넷 보안에서의 신원인증(Authentication)에 해당합니다.

지금까지 비대칭키와 대칭키를 사용한 암호화 및 서명 등에 관한 내용을 전해드렸습니다. 이후의 연재도 SSL/TLS를 설명하기 위한 주제들로 이어가겠습니다.

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.