SHA-1

암호학에서, SHA-1(Secure Hash Algorithm 1)은 깨졌지만 현재까지 널리 쓰이는 입력 값을 받고 메시지 요약문(일반적으로 16진수 40자리)으로 알려진 160비트(20바이트)의 해시 값을 생성하는 해시함수이다. 미국 국가안보국에 의해 설계되었으며, 미국 연방 정보 처리 표준이다.

2005년부터 SHA-1은 충분한 재원이 있는 공격자에게 안전하지 않다고 여겨진다. (떄문에 2010년부터 많은 조직들이 이것의 대체를 권하고 있다.) NIST는 2011년 SJA-1의 사용을 공식적으로 반대하고 2013년 디지털 서명을 위한 사용을 금지했다. 2020년을 기준으로 SHA-1에 대항하는 선택 접두어 공격(chosen-prefix-attacks)이 현실화되었다. 이러한 이유로 가능한 빨리 SHA-1을 제거하고 SHA-2나 SHA3를 사용할 것이 권고된다. 디지털 서명을 위해 사용되는 환경에서는 SHA-1를 제거하는 것이 시급하다.

모든 주요 웹 브라우저 벤더들은 2017년 SHA-1 SSL 인증서의 수용을 중단했다. 2017년 2월, CWI 암스테르담과 구글은 SHA-1에 대한 충돌 공격을 수행했다고 발표했으며 동일한 SHA-1 해시를 생성하는 구별된 2개의 PDF 파일들을 출판했다. 그러나 SHA-1은 여전히 HMAC에 대해서는 안전하다.

마이크로소프트는 2020년 8월 7일 윈도우 업데이트에 SHA-1 코드 서명 지원을 중단했다.

1. 개발과정

SHA-1은 MIT의 Ronald L. Rivest가 디자인한 MD2, MD4, MD5 메시지 다이제스트 알고리즘의 설계에서 사용한 것과 유사한 원리에 기반한 메시지 다이제스트를 생성하지만, 더 큰 해시 값(128비트보다 큰 160비트)을 생성한다.

SHA-1은 미국 정부의 캡스톤 프로젝트의 일환으로 개발되었다. 이 알고리즘의 원래 설명서는 1993년 미국 정부 표준 기관 NIST(National Institute of Standards and Technology)에 의해 Secure Hash Standard, FIPS PUB 180이라는 제목으로 발표되었다. 이 버전은 종종 SHA-0이라고도 불린다. SHA-0는 1995년 NSA에 의해 발표 이후 빠르게 철수되었으며, 1995년 FIPS PUB 180-1에서 공개된 일반적으로 SHA-1이라고 불리는 수정된 버전으로 대체되었다. SHA-1은 SHA-0와 암축함수에 비트 회전 연산을 하나 추가했다는 점이 다르다. NSA에 따르면, 이는 암호학적 보안을 감소시키는 SHA-0의 문제점을 보완하기 위해 이루어졌다고 밝혔지만, 그들은 더 이상의 설명을 제공하지는 않았다. 일반적으로 SHA-1은 SHA-0에 비해 암호학적 공격이 힘들다고 알려져 있다.

2. 응용 프로그램

2.1. 암호화학에서

SHA-1은 TLS와 SSL, PGP, SSH, S/MINE, IPsec을 포함하여 널리 사용되는 보안 응용 프로그램과 프로토콜의 일부를 형성한다. 이러한 애플리케이션은 MD5를 사용할 수도 있다. (MD5와 SHA-1은 모두 MD4에서 파생되었다.)

SHA-1과 SHA2는 기밀이 아닌 민감한 정보를 보호하는 특정 미국 정부 응용 프로그램(다른 암호화 알고리즘 및 프로토콜안에서 사용을 포함)에서 사용하기 위한 법률에서 요구되는 해시 알고리즘이다. 또한 FIPS PUB 180-1은 민간 및 상업 기관에서 SHA-1을 채택하고 사용하는 것을 권장했다. SHA-1은 대부분의 행정적 사용에서 퇴역하고 있다; 미국 NST는 “연방 기관은 가능한 빨리 충돌 저항성이 부족한 애플리케이션...의 SHA-1의 사용을 중단해야하며, 2010년 이후에는 이러한 애플리케이션에 SHA-2를 사용해야만 한다.”라고 말했지만, 나중에 SHA-1이 오래된 디지털 서명과 타임 스탬프를 확인하는데 사용될 수 있도록 완화되었다.

SHA는 디지털 서명 표준(Digital Signature Standard)때문에 만들어졌다.

SHA 해시 함수는 SHACAL 블록 암호의 기초로 사용되었다.

2.2. 데이터 무결성

Git, Mercurial, Monotone과 같은 리비전 관리 시스템은 보안을 위한 것이 아닌, 리비전을 식별하고 돌발적인 변형으로 인해 데이터가 변경되지 않도록 SHA-1을 사용한다. Linus Torvalds는 깃에 대해 다음과 같이 말했다:

만약 디스크 손상, DRAM 손상, 혹은 어떤 문제라도 있다면, Git은 알아차릴 것이다. 보장이 되냐의 문제가 아니다. 당신은 악의적인 생각을 가진 사람을 만날 수 있다. 그들은 성공하지 못할것이다.[…] 아무도 SHA-1을 깨지 못했지만, Git에 관한 한 SHA-1은 보안적인 기능을 수행하지는 않는다. 순전히 일관성 검사만 한다. 보안 부분은 다른 곳에 있다. 많은 사람들은 Git이 SHA01을 사용하고 SHA-1은 암호학적으로 안전을 위해 사용되기 때문에 SHA-1이 보안 기능이라고 생각한다. 보안과는 전혀 상관없다. 당신이 얻을 수 있는 것은 최고의 해시일 뿐이다.

장담하건데, 만약 Git에 데이터를 저장하면, 5년 후 하드디스크에서 DVD로 변환되어 새로운 기술로 복사한 후, 5년 후에 다시 출력되는 데이터가 입력한 데이터와 정확히 동일한지 확인할 수 있다.

내가 커널을 신경 쓰는 이유 중 하나는, 우리가 사람들이 커널 소스 코드 저장소를 손상시키려고 했던 비트키파 사이트 중 하나에 침입했기 때문이다.

그러나 Git은 공격자가 은밀하게 중복된 파일을 만드는 것을 예방하는 보안 기능으로 SHA-1의 제2 역상 공격을 필요하지 않는데, 충돌시 항상 초기 버전을 유지하는 것을 선호하기 때문이다.

3. 암호 해독과 검증

L이 메시지 다이제스트의 비트 수인 해시함수의 경우, 주어진 메시지 다이제스트에 해당하는 메시지를 찾는 것은 항상 거의 2^L개의 평가에서 brute force 검색을 사용하여 수행될 수 있다. 이를 역상 공격이라고 하며 L과 특정 컴퓨팅 환경에 따라 실용적인지 아닐지가 달려있다. 그러나 동일한 메시지 다이제스트를 생성하는 두 개의 다른 메시지를 찾는 것으로 구성된 충돌은 생일 공격을 사용하여 평균적으로 약 1.2\*2^(L/2)개의 평가만 필요로 한다. 따라서 해시 함수의 강도는 일반적으로 메시지 다이제스트 길이의 절반인 대칭 암호와 비교된다. 160비트 메시지 다이제스트가 있는 SHA-1은 원래 80비트 강도로 생각되었다.

패스워드 저장소와 같이 암호화 해시를 사용하는 일부 응용 프로그램은 충돌 공격의 영향을 최소화한다. 지정된 계정에 대해 작동하는 패스워드를 만들기 위해서는, 원래 패스워드의 해시에 접근하는 것뿐만 아니라 역상 공격이 필요해야 하며, 이는 사소한 것일 수도 그렇지 않을 수도 있다. 패스워드 암호화를 되돌리는 것(예: 다른 곳에서 사용자 계정에 맞는 암호를 얻는 과정)은 공격으로 가능해지지는 않는다. (그러나 안전한 패스워드 해시조차도 약한 암호에 대한 brute-force 공격은 막을 수 없다.)

문서 서명의 경우, 공격자는 기존 문서에서 서명을 위조할 수는 없다 : 공격자는 위험하지 않은 문서와 유해한 문서 한쌍을 준비하고, 개인키를 가진 사람이 무해한 문서에 서명을 하도록 해야 한다. 이것이 실제로 가능한 상황이 있다; 2008년 말까지 MD5 충돌을 이용하여 위조된 SSL 인증서를 만드는 것이 가능했다.

알고리즘의 블록과 반복 구조 및 추가 최종 단계의 부재로 인해, 모든 SHA함수(SHA-3 제외)는 길이 확장 및 메시지 충돌 공격에 취약하다. 이러한 공격을 통해 공격자는 키를 모르는 상태에서 메시지를 확장하고 해시를 재계산하여 키 해시(SHA(메시지||키) 또는 SHA(키||메시지))만으로 서명된 메시지를 위조할 수 있다. 이러한 공격을 방지하기 위한 간단한 개선법은 해시를 2번 계산하는 것이다: (0 블록 0^b의 길이는 해시함수의 블록 사이즈와 같다.)

3.1. SHA-0

CRYPTO 98에서 두명의 프랑스 연구자 Florent Chabaud and Antoine Joux는 SHA-1에 대한 공격법을 제시했다: 충돌은 같은 크기의 이상적인 해시 함수인 2^80보다 적은 2^61에서 나타날 수 있다.

2004년, Biham 과 Chen은 (SHA-0 충돌)거의 같은 해시 값을 가진 2개의 메시지를 발견했다; 이 상황에서 160비트 중 142비트는 동일했다. 그들은 또한 SHA-0의 완전한 충돌이 80라운드 중 62 라운드로 감소했다는 것을 발견했다.

이후 2004년 8월 12일, Joux, Carribault, Lemuet, and Jalby는 SHA-0 알고리즘의 완전한 충돌을 발표했다. 이것은 Chabaud와 Joux 공격의 일반화를 사용하여 수행되었다. 발견된 충돌에서는 2^51의 복잡성이 있었고 256개의 Litanium 프로세서가 장착된 슈퍼컴퓨터에서 약 80000 프로세서 시간이 소유되었다.(컴퓨터로 13일을 사용한 경우와 동일)

2004년 8월 17일, CRYPTO 2004의 Rump 세션에서 Wang, Feng, Lai, and Yu에 의해 MD5, SHA-0및 기타 해시 함수에 대한 예비 조사 결과가 발표되었다. SHA-0에 대한 그들의 공격법의 복잡성은 2^40으로, Joux 등의 공격보다 훨씬 우수하다.

2005년 2월, Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu가 2^39번 작동하는 SHA-0에서 충돌을 발견할 수 있는 공격을 발표했다.

부메랑 공격을 적용한 2008년의 또다른 공격은 충돌 발견의 복잡성을 2^33.6으로 낮추었는데, 이는 2008년부터 일반적인 PC에서 1시간정도 걸리는 것으로 추정된다.

SHA-0에 대한 결과에 비추어 일부 전문가들은 새로운 암호 시스템에서 SHA-1을 사용하는 계획을 수정해야 한다고 제안했다. CRYPTO 2004 결과가 발표된 이후, NIST는 SHA-2의 사용을 2010년까지 단계적으로 중단할 계획이라고 밝혔다.

3.2. 공격

2005년 초, Vincent Rijmen과 Elisabeth Oswald는 SHA-1의 축소판(80라운드 중 53라운드)에 대한 공격을 발표했는데 이는 280개 미만의 연산을 통해 충돌을 찾아낸다.

2005년 2월, Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu의 공격법이 발표되었다. 이 공격은 269겨 미만의 연산이 필요한 SHA-1의 모든 버전에서 충돌을 찾을 수 있다. (brute-force 검색에는 280개의 연산이 필요하다.)

Marc Stevens가 만든 공격 중 하나는 클라우드 서버에서 CPU 파워를 이용해 단일 해시 값을 깨는데 약 277만 달러(2012년 기준)의 비용이 드는 공격이다. Stevens는 이 공격을 HashClash라는 프로젝트에서 개발하여, 차등경로공격을 구현했다. 2010년 11월 8일, 그는 2^57.5 SHA-1 압축과 동등한 복잡성으로 작동하는 완전한 SHA-1에 충돌 공격을 가했다고 주장했다. 그는 이 공격이 2^61 복잡성과 완전한 충돌로 확장할 수 있다고 추정했다.