현대암호학의 태동 [4]: 리베스트, 샤미르, 애들먼의 RSA 공개키 암호 스킴

Ethereum Foundation 이기우 연구원

스물아홉의 한 남자가 소파에 누워 있다. 자정에 가까운 시간 와인에 조금 취한 채 천장을 멍하니 바라보다가 이내 수학책을 뒤적인다. 책장을 넘기다 멈추고 다시 허공을 바라본다. 몇 번이나 반복했을까, 불현듯 하나의 생각이 스친다. 안개가 걷히듯 머릿속이 맑아진다. 단순하면서도 우아하다. 그리고 이상하리만치 확신이 든다. 이것이 작년부터 붙잡고 씨름해 온 질문에 대한 해답일 것이라는 강한 느낌이 든다. 그는 곧바로 소파에서 일어나 타자기 앞에 앉는다. 밤새도록 손을 멈추지 않는다. 동이 트기 전 사실상 논문 한 편이 완성된다. 1977 년 4 월, 역사상 최초의 공개키 암호, RSA 가 탄생하는 순간이다.

작년, MIT 의 젊은 교수 리베스트(Ron Rivest; 1947~)는 막 발표된 디피와 헬만의 논문[참고문헌 6]을 읽고 깊이 매료되었다. 공개키 암호라는 개념, 그리고 트랩도어 일방향 함수라는 아이디어는 그가 오랫동안 그려오던 이상과 정확히 맞아떨어졌다. 이론적으로는 우아하면서도 현실 세계에 큰 영향을 미칠 수 있는 주제였기 때문이다. 이전 화에서 살펴본 것처럼, 디피와 헬만은 트랩도어 일방향 함수를 통해 공개키 암호를 설계하는 청사진을 제시했지만, 이를 실제로 구현하는 단계에는 이르지 못했다. 리베스트는 바로 이 지점, 디피와 헬만이 남긴 과제, 즉 트랩도어 일방향 함수를 구체적으로 설계해 진정한 의미의 공개키 암호화 스킴을 구현하는 문제에 강하게 끌리게 된다.

Figure 1 리베스트(Ron Rivest; 1947~)

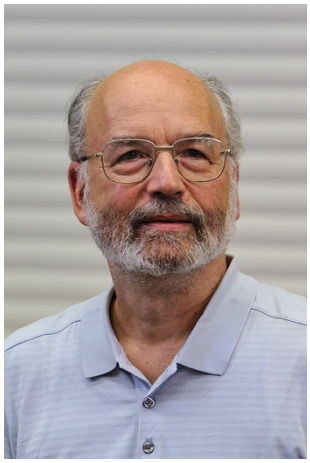

Figure 2 샤미르(Adi Shamir; 1952~)

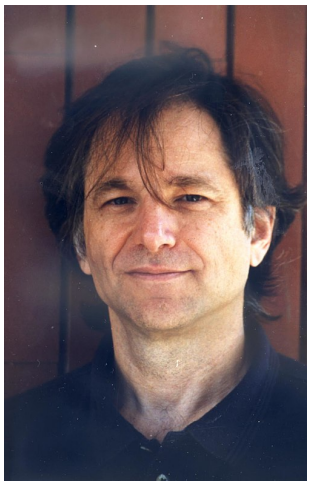

Figure 3 애들먼(Leonard Adleman; 1945~)

리베스트는 이 문제를 혼자서만 붙잡고 씨름하지 않았다. 그는 MIT 에서 같은 층에 연구실을 쓰던 샤미르(Adi Shamir; 1952~)와 애들먼(Leonard Adleman; 1945~)에게 이 문제를 소개했고, 세 사람은 곧 공동연구에 들어갔다. 공개키 암호화가 과연 가능한 것인지, 아니면 애초에 불가능한 목표인지조차 확신할 수 없는 상태에서 수개월에 걸친 연구가 이어졌다. 리베스트와 샤미르가 새로운 아이디어를 제시하면, 애들먼은 그 안전성을 집요하게 검토하며 수차례 이를 폐기시켰다. 리베스트는 훗날 이 시기를 회상하며, 어느 순간에는 차라리 불가능함을 증명하는 편이 더 현실적으로 느껴질 정도였다고 말한다. 그만큼 문제는 쉽게 풀리지 않았고, 여러 접근은 번번이 막다른 길에 부딪혔다. 그럼에도 불구하고 세 사람은 포기하지 않았다. 그리고 어느 봄밤, 그들의 집요한 연구는 리베스트의 머릿속에서 하나의 단순하면서도 우아한 형태로 마침내 응축된다.

날이 밝자, 리베스트는 여느 때와 다름없이 학교로 향한다. 그리고 샤미르와 애들먼을 찾아가 밤새 써 내려간 논문 초안을 건넨다. 애들먼은 늘 그랬듯이 원고를 받아 들고 꼼꼼하게 안전성을 검토한다. 달라진 것은 그 검토의 결론이었다. 애들먼은 끝내 공격을 찾지 못했고, 마침내 공개키 암호의 설계에 성공했을지도 모른다는 사실을 처음으로 인정한다.

그러나 이어진 말은 의외였다. 이론전산학의 관행에 따라 저자 이름이 알파벳순으로 애들먼–리베스트–샤미르가 나란히 적힌 논문 초안을 가리키며, 애들먼은 자신의 이름을 논문에서 빼달라고 요청한다. 이 아이디어는 결국 리베스트의 것이며, 본인 생각에 자신이 저자로 들어갈 만한 기여를 하지 않았다는 이유에서였다. 리베스트는 몇 달에 걸친 세 사람의 공동 연구가 없었다면 이 아이디어에 도달하지 못했을 것이라며 애들먼을 설득했다. 결국 애들먼은 "마지막 저자로 들어간다"라는 조건으로 공동 저자가 되는 데 동의한다. 그렇게 알파벳순으로라면 ARS 가 되었을 최초의 공개키 암호는, 논문에 적힌 순서대로 저자들의 머리글자를 따 RSA 라고 불리게 된다. 애들먼은 훗날 이 논문을 두고, 순수수학자로서의 정체성이 강했던 당시에는 본인이 저자로 이름을 올릴 논문들 가운데 가장 시시한 것이 될 것이라고 생각했다고 회고한다. 그러나 세상의 평가는 정반대였다. 완성된 논문 <A Method for Obtaining Digital Signatures and Public-Key Cryptosystems(전자서명과 공개키 암호시스템을 구현하는 방법)>[참고문헌 1]은 애들먼의 이름을, 그리고 리베스트와 샤미르의 이름을, 전산학의 역사에 남기게 된다.

RSA 공개키 암호 스킴

RSA 공개키 암호 스킴을 이해하기 위해, 먼저 트랩도어 일방향 함수를 다시 떠올려보자. 트랩도어 일방향 함수란 한쪽 방향으로의 계산은 매우 쉽지만, 그 결과만으로는 입력을 되돌리는 것이 극도로 어려운 함수이며, 오직 특정한 비밀 정보, 즉 트랩도어를 알고 있을 때만 효율적으로 역산이 가능한 수학적 구조를 말한다. 핵심은 정방향 연산과 역방향 연산 사이의 비대칭성, 그리고 트랩도어를 알고 있는 사람과 그렇지 않은 사람

사이의 비대칭성에 있다. 이러한 비대칭성이 바로 공개키 암호를 가능하게 한다. 일방향 함수의 정의(description) 자체를 공개키로, 트랩도어 정보를 비밀키로 삼고, 함수를 계산하는 과정을 암호화에, 역함수 계산을 복호화에 대응시키면 자연스럽게 공개키 암호화 스킴이 구성된다.

관심 있는 독자라면 한 번쯤 들어봤겠지만, RSA 스킴의 핵심 재료는 소인수분해이다. 두 개의 큰 소수 p, q를 골라 이를 곱해 하나의 정수 $N = pq$를 계산하는 일은 매우 쉽다. 반면, 그 결과인 N 만 주어졌을 때 다시 p 와 q 를 찾아내는 소인수분해는 전혀 다른 차원의 복잡도를 가진다. 고대 그리스의 에라토스테네스부터 천하의 가우스를 거쳐 현대에 이르기까지 수많은 수학자들이 소인수분해를 연구해 왔지만, 오늘날 가장 빠른 알고리즘[1]조차도 충분히 큰 입력 앞에서는 무력해진다.[2]

소인수분해는 분명 이러한 비대칭성을 제공하는 매력적인 재료이지만, 이것만으로 곧바로 트랩도어 일방향 함수를 설계하기 쉬운 것은 전혀 아니다. 실제로 디피와 헬만 역시 한때 소인수분해를 기반으로 공개키 암호를 설계할 수는 없을지 고민했던 것으로 알려져 있다. 그러나 그들은 소인수분해를 암호화 연산과 직접적으로 연결하는 데에는 이르지 못했다.

리베스트가 오랜 연구 끝에 도달한 해법은 $N = pq$ 에서 나타나는 곱셈과 소인수분해의 비대칭성 위에, 또 하나의 비대칭성, 즉 거듭제곱 연산과 그 역연산인 거듭제곱근 계산 사이의 비대칭성을 사용하는 것이다. 구체적으로 RSA 스킴은 두 개의 큰 소수 p, q 를 비밀키(트랩도어)로 두고 둘의 곱 $N = pq$ 과 지수 e 를 공개키로 한다.[3] 이때, 평문 m 에 대해 암호문 c 는 아래와 같이 N 을 법으로 e -거듭제곱을 통해 계산된다.

$$c = m^e \pmod{N}$$

이 연산은 N 과 e 만 알고 있으면 누구나 효율적으로 계산할 수 있다. 반면, 암호문 c 로부터 원래 메시지 m 을 복원하려면, 즉 법 N 에서 c 의 e -제곱근을 계산하려면 상황이 전혀 달라진다. N 의 소인수분해 결과를 알지 못할 때, 이러한 역연산을 효율적으로 수행하는 방법은 알려져 있지 않다.

---

[1] 오늘날 가장 빠른 알고리즘인 General Number Field Sieve 알고리즘도 N -비트 정수를 소인수분해 하는 데에 $\exp\left(\tilde{O}(\log^{1/3} N)\right)$의 시간복잡도를 갖는다.[참고문헌 5]

[2] 이는 고전적인 의미의 알고리즘에 한정했을 때의 이야기이며, 언제 등장할 지 모르는 양자컴퓨터까지 고려하면 상황은 달라진다. 쇼어(Peter Shor, 1959~)는 1994 년 소인수분해를 효율적으로, 즉 다항시간 안에 해결할 수 있는 "양자 알고리즘"을 제안했다. 이에 관심 있는 독자는 김한영 교수님의 Horizon 글 〈양자 알고리즘: 소인수 분해 알고리즘〉을 참고해도 좋겠다. 한편, 장차 양자컴퓨터가 등장하더라도 안전할 것으로 기대되는 양자내성암호(post-quantum cryptography)는 21 세기 암호학의 중요한 담론 중 하나다.

[3] 이때, e 는 N의 오일러 함숫값 $\phi(N)$ 과 서로소여야 한다.

그러나 비밀키(트랩도어) p, q 를 알고 있는 경우에는 상황이 또다시 달라진다. 오일러의 정리를 이용하면 e $-제곱근을 효율적으로 계산할 수 있기 때문이다. 실제로 $N = pq$ 라는 구조 덕분에 우리는 N 의 오일러 함숫값 $\phi(N) = (p-1)(q-1)$ 을 즉시 계산할 수 있으며, 공개 지수 e 에 대해 $d = e^{-1} \pmod{\phi(N)}$ 을 만족하는 역원 d 역시 유클리드 알고리즘을 통해 빠르게 구할 수 있다. 이렇게 얻은 d 를 사용하면 암호문 c 는 아래와 같이 N 을 법으로 d-거듭제곱을 통해 복호화할 수 있다.

$$m = c^d \pmod{N}$$

이는 $ed = 1 \pmod{\phi(N)}$일 때 오일러의 정리에 의해 $c^d = m^{ed} = m \pmod{N}$이 성립하기 때문이다. 요컨대 N의 소인수분해 정보는 RSA 에서 e-제곱근 연산을 가능하게 하는 트랩도어로 작용한다.

RSA 스킴의 안전성

RSA 스킴의 안전성은 흔히 소인수분해의 어려움에 기반한다고 설명된다. 하지만 이는 엄밀하게 따지고 들면 정확한 표현은 아니다. 물론 소인수분해를 쉽게 풀 수 있다면 RSA 는 즉시 붕괴된다. 그러나 이는 가능한 공격 경로 중 하나에 불과하다. 소인수분해를 직접 수행하지 않고도 RSA 암호문을 복호화하는 전혀 다른 방법이 존재할 가능성을 배제할 수 없다. 소인수분해를 풀 수 있을 때 RSA 가 무력화된다는 사실은, 단지 RSA 스킴을 공격하는 것이 소인수분해보다 적어도 어렵지 않다는 점을 말해줄 뿐, RSA 스킴의 안전성을 뒷받침하는 근거가 되지는 않는다.

오히려 암호학적으로 중요한 질문은 반대방향이다. 즉, 누군가 RSA 스킴을 효율적으로 공격할 수 있다면, 그는 반드시 소인수분해 문제도 효율적으로 풀 수 있음을 보일 수 있는가라는 질문이다. 이러한 관계가 성립할 때야 비로소 우리는 RSA 스킴을 공격하는 것이 소인수분해만큼이나 어렵다고, 다시말해 RSA 의 안전성이 소인수분해에 기반한다고 말할 수 있을 것이다. 그러나 안타깝게도 오늘날까지 RSA 의 안전성을 이처럼 소인수분해 문제에 온전히 기반시킬 수 있는지는 알려져 있지 않다.[4][참고문헌 8]

이러한 이유로 암호학에서는 RSA 의 안전성을 보다 직접적인 계산 문제로 정식화한다. 바로 RSA 문제이다. 이는 두 큰 소수 p, q 의 곱으로 이루어진 공개키 N 이 소인수분해 정보 없이 주어졌을 때, 법 N에서 주어진 암호문으로부터 e-제곱근을 계산하는 문제를 말한다. 앞서 살펴본 것처럼 RSA 문제는 소인수분해

[4] 같은 이유로, 저번 화에서 살펴본 디피-헬만 키교환 프로토콜이 이산로그 문제에 기반한다고 말하는 것 역시 엄밀히는 부정확한 표현이다.

문제보다 더 쉽다. 그렇다면 우리는 이 문제가 어렵다고, 다시 말해 RSA 스킴이 안전하다고 어떻게 주장할 수 있을까? 결국 소인수분해 문제와 마찬가지로, 오랜 시간에 걸쳐 형성된 일종의 사회적 합의에 가깝다. RSA 가 제안된 이후 반세기에 가까운 시간 동안 수많은 암호분석가가 RSA 문제를 직접적으로 공략하려 시도해 왔지만, 오늘날까지도 소인수분해를 이용한 직접적인 공격을 제외하고는 실질적인 공격이 발견되지 않았다. 오히려 시간은 RSA 문제의 완강함을 반복해서 증명해 왔다.

물론 RSA 문제의 구체적인 난이도에 대한 평가는 계산 능력과 알고리즘의 발전과 함께 계속해서 변화해 왔다. 예컨대 1977 년, 마틴 가드너(Martin Gardner; 1914–2010)는 Scientific American 의 Mathematical Games 칼럼에 "A new kind of cipher that would take millions of years to break(해독에 수백만 년이 걸리는 새로운 암호 방식)"이라는 제목의 글을 실으며 RSA 암호를 소개했다.[참고문헌 9] 이 글에는 하나의 RSA 암호문이 예제로 제시되었다. 십진수로 각각 64 자리와 65 자리인 두 소수의 곱으로 이루어진 129 자리 정수 N, 공개 지수 e, 그리고 암호문 c가 다음과 같이 주어졌다.

$$N = 114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958$$

$$e = 9007$$

$$c = 199935131497805100452317122740260647423204017058391463103703717406259716089489275043099209626725$$

당시 RSA 의 세 설계자는 이 암호문을 복호화하는 데에 우주 나이를 아득히 넘는 4 경(京)년이 걸릴 것이라 예상하며, 이를 풀어내는 사람에게 100 달러를 지급하겠다고 적어 두었다. 이 문제는 이후 RSA-129 라는 이름으로 불리며, 하나의 이정표처럼 여겨지게 되었다.

그러나 계산 환경과 알고리즘은 예상보다 훨씬 빠르게 발전했다. 결국 RSA-129 는 1994 년에 이르러 약 8 개월 동안, 전 세계 약 600 명의 자원봉사자가 제공한 1600 여 대의 컴퓨터 자원을 동원한 분산 계산을 통해 해결되었다.[참고문헌 10] 이는 "100 달러를 버는 가장 비싼 방법"이라는 우스갯소리를 남겼지만, 동시에 암호 스킴의 안전성 평가가 시간과 기술의 발전에 따라 얼마나 크게 달라질 수 있는지를 분명히 보여주었다. 더 나아가 2015 년에는 클라우드 컴퓨팅 환경을 이용해 약 30 달러의 비용으로, 하루 남짓한 시간 안에 RSA-129 를 푸는 것이 가능해졌다.[참고문헌 11] 관심 있는 독자라면 직접 이를 실험해 보는 것도 충분히 가능하다. 참고로 오늘날에는 충분한 안전성을 위해 주로 RSA-2048 이 사용된다. 이는 각각 1024 비트 길이의 두 소수의 곱으로 이루어진 2048 비트 정수, 즉 십진수로 약 617 자리인 정수 N을 사용한다는 뜻이다.

RSA 설계 이후 실용화까지

RSA 세 사람은 2002 년, 전산학의 노벨상이라 불리는 튜링상을 수상했다. 흥미롭게도 이는 2015 년에 수상한 디피와 헬만보다 훨씬 이른 시기에 공로를 인정받은 것이다. 수상 공적은 다음과 같다.

> 공개키 암호를 실용 기술로 정착시키는 데에 대한 탁월한 기여 (for their ingenious contribution to making public-key cryptography useful in practice)

주목할 만한 점은 수상 공적이 단순히 "최초의 공개키 암호 설계"에 그치지 않는다는 것이다. 이는 세 사람이 1976 년 최초의 설계 이후에도 공개키 암호가 실용화되기까지 많은 시행착오와 고군분투를 겪었음을 엿볼 수 있는 대목이다.

가장 먼저 짚고 넘어가야 할 점은 1970 년대의 컴퓨팅 환경이 오늘날의 기준으로는 상상하기 어려울 정도로 느렸다는 사실이다. RSA 스킴의 핵심인 큰 소수를 찾는 것은 말할 것도 없고[5], RSA 암복호화 연산 역시 당시의 대형 연구용 컴퓨터 환경에서도 결코 가벼운 작업이 아니었다. 이러한 성능 한계를 인식한 셋은 알고리즘 제안에 그치지 않고, 당시로서는 최첨단이던 VLSI[6] 기술을 활용한 전용 하드웨어 구현까지 적극적으로 탐구했다. 그로부터 수십 년이 흐른 오늘날, 한때 대형 컴퓨터에서나 가능하던 공개키 암호 연산은 컴퓨터 시스템 성능의 지수적인 향상, 즉 무어의 법칙(Moore's Law)으로 요약되는 기술적 진보에 힘입어 모바일 환경에서도 밀리초 단위로 수행될 수 있게 되었다.

RSA 의 실용화 과정에는 여러 비기술적 어려움도 존재했다. RSA 가 제안된 1970 년대 후반은 인터넷의 기초 인프라는 이미 갖추어져 있었지만, 오늘날 우리에게 익숙한 월드 와이드 웹(WWW)이 등장하기 이전이었다. 다시 말해, 공개키 암호가 즉각적으로 활용될 만한 대중적·상업적 시장은 사실상 존재하지 않았다. 암호 기술의 필요성을 실질적으로 인식하던 수요자는 정부와 군, 일부 대형 기관에 한정되어 있었고, 일반 사용자와의 접점은 극히 제한적이었다. 이러한 환경 속에서 리베스트, 샤미르, 애들먼은 RSA 알고리즘의 보급과 상용화를 목표로 1982 년 회사 RSA Data Security 를 설립하고 시장 개척에 나섰다. 그러나 전자상거래를 비롯한 웹 기반

---

[5] 리베스트는 훗날 회고에서[참고문헌 3], 당시 큰 소수를 생성하는 일이 얼마나 어려웠는지를 설명하며, 어떤 사업가가 큰 소수를 대신 찾아서 판매하겠다는 비즈니스 아이디어를 내놓을 정도였다고 전한다. 물론 다른 사람이 생성해 준 비밀키를 사용하는 것은 암호학적으로 전혀 말이 되지 않는 발상이지만.

[6] Very Large Scale Integration

서비스가 본격적으로 등장하기 전까지, 공개키 암호가 창출할 수 있는 상업적 가치는 제한적일 수밖에 없었다. 결과적으로 공개키 암호의 본격적인 시장은 1990 년대 초반이 되어서야 웹의 확산과 함께 비로소 형성되었다.

오늘날의 시점에서 보면 의외로 느껴질 수 있지만, RSA 의 실용화 과정에서 기술적·상업적 어려움 못지않게 큰 장애물로 작용한 것은 정부 차원의 규제였다. 미국의 수출 통제 체계는 강력한 암호 기술을 군사 기술의 일종으로 취급하며 해외 반출을 엄격히 제한했고, 그 결과 암호 기술의 공개와 보급은 오랫동안 정치적 논쟁의 대상이 되었다.이러한 규제 분위기는 RSA 논문의 공개 과정에도 직접적인 영향을 미쳤다. 당시 미국 정보기관 NSA 는 암호 연구의 공개에 대해 지속적으로 우려를 표명하고 있었으며, 미국 정부 산하의 과학 연구 지원 기관인 NSF 의 암호 관련 연구 지원이나 IEEE 등의 학술 단체의 암호 관련 활동에도 압력을 작용하던 분위기였다. 이에 따라 리베스트, 샤미르, 애들먼은 RSA 알고리즘을 발견한 직후 이를 곧바로 발표하지 못하고, MIT 법률팀과 함께 법적 문제의 소지를 신중히 검토한 뒤에야 논문을 공개할 수 있었다. 정부 규제는 이후 RSA 의 기업 활동에도 실질적인 제약으로 이어졌다. 소프트웨어가 국경을 넘나들기 시작한 시점에도 규제 체계는 여전히 물리적 무기 수출을 전제로 설계되어 있었다. 그 결과 동일한 제품을 국내용과 해외용으로 구분해 배포해야 했으며, 해외용 버전에는 키 길이를 제한한 이른바, '약화한 암호'를 적용할 수밖에 없었다. 이러한 RSA 의 실용화를 향한 고군분투는 스티븐 레비의 저서 [참고문헌 5]에 생생하게 기록되어 있으니, 관심 있는 독자라면 일독을 권한다.

Figure 5 1990 년대 미국의 암호 수출 규제를 풍자하기 위해 제작된RSA 티셔츠. 티셔츠에는 RSA 스킴의 구현 코드가 바코드 형태로 인쇄되어 있었으며, 이로 인해 군수 물자로 분류되어 해외 수출이 금지되었고, 미국 국민이 이를 외국인에게 보여주는 것조차 위법으로 간주되었다.

Figure 4 회사 RSA Data Security 의 로고

RSA 스킴의 안전성 되짚어보기

그런데... 소인수분해를 넘어 RSA 문제가 어렵다면 RSA 스킴은 정말로 안전하다고 말할 수 있을까? 필자는 꼭 그렇지 않을 수도 있다는 것을 설명해 보려 한다. 바로 앞에서 RSA 스킴의 성공과 상용화를 이야기했기 때문에 다소 뜬금없게 느껴질 수도 있겠지만, 조금만 참고 따라와 주시길.

가장 먼저 짚어야 할 점은, 위에서 묘사한 RSA 암호화 방식이 무작위성(randomness)이 전혀 없는 결정적(deterministic)이라는 사실이다. 즉, 같은 평문 m 을 같은 공개키 (N, e) 로 암호화하면 언제나 동일한 암호문 $m^e \pmod N$ 이 생성된다. 이 특성은 직관적으로는 별문제가 없어 보일 수 있지만, 실제 환경에서는 심각한 취약점으로 이어질 수 있다.

예를 들어 주민등록번호처럼 가능한 값의 범위가 비교적 제한된 정보를 RSA 로 암호화한다고 가정해 보자. 공격자는 공개키를 이용해 가능한 모든 주민등록번호 후보를 직접 암호화한 뒤, 그 결과를 관측된 암호문과 비교할 수 있다. 주민등록번호의 가능한 가짓수는 대략 2^{40} 수준에 불과하며, 이는 GPU 나 ASIC 과 같은 하드웨어 가속을 활용할 경우 현실적으로 탐색 가능한 범위에 해당한다. 이 과정에는 소인수분해도, 비밀키도 필요하지 않다. 단지 "암호화가 결정적이다"라는 사실만으로 무차별 대입 공격이 가능해지는 것이다. RSA 가 일방향 트랩도어 함수를 사용하고 있기 때문에 공개키만으로 비밀키를 복구할 수는 없지만, 메시지 공간이 작을 경우에는 비밀키를 우회해 평문을 직접 복구하는 경로가 존재하는 셈이다. 다시말해, RSA 는 일방향 트랩도어 함수를 기반으로 하고 있기 때문에 공개키만으로 비밀키를 복구하기는 어렵지만, 메시지 공간이 충분히 작다면, 비밀키를 우회해 평문을 직접 복구하는 경로가 존재하게 된다는 것이다.

이러한 이유로, 논문에 제시된 그대로의 RSA 암호화 방식은 오늘날 textbook RSA 라 불리며 실제 시스템에서는 사용되지 않는다. 대신 암호화 과정에 무작위성을 도입하기 위해 랜덤한 패딩을 덧붙이는 방식이 사용된다. 대표적으로 OAEP 와 같은 패딩 기법은 동일한 평문이라도 매번 서로 다른 암호문이 생성되도록 만들어, 앞서 살펴본 결정적 암호화의 취약점에 대응한다.

그렇다면 textbook RSA 는 안전하다고 할 수 없는 걸까? 그럼 랜덤한 패딩을 적용한 RSA 는 안전하다고 말할 수 있을까? 궁극적으로, 우리가 오늘날 안전하다고 믿는 이러한 구성 방식이 앞으로도 새로운 공격으로부터 안전하다는 것을 어떻게 확신할 수 있을까?

증명가능한 안전성을 향해

그런데 왜 이런 모호함이 생기는 걸까? 이는 우리가 섀넌의 정보이론적 안전성을 살펴본 이후로 애써 외면해 온 질문과 맞닿아 있다. 바로 안전성의 정의다. 메시지 길이만큼 비밀키가 길어야 하는 정보이론적 안전성의 한계를, 소인수분해가 어렵다는 가정으로 우회하고 있는 만큼 RSA 는 정보이론적 안전성을 만족하지 않는다. 그렇다면 우리가 지금껏 RSA 를 다루며 이야기해 온 '안전성'은 정확히 무엇이란 말인가? 그리고 그 '안전성'은 과연 모두가 납득할 만한 정의를 갖고 있는가?

우리가 안전한 암호 스킴에게 무엇을 기대하는지 명확하지 않다면, RSA 의 안전성은 증명은 고사하고 반증조차 가능하지 않다.[7] 우리에게는 이런 식으로 표류하는 개념들을 단단히 붙들어 둘 이론적 기반이 필요하다. 안전성에 대한 엄밀한 정의를 갖춘 뒤에야 암호학은 반증 가능한 명제들을 다룰 수 있고, 그제서야 비로소 우리는 암호학을 과학이라 부를 수 있을 것이다.

다음글에서는 골드와서(Shafi Goldwasser; 1958~)와 미칼리(Silvio Micali; 1954~)가 "증명가능한 안전성(Provable Security)"을 바탕으로 암호학을 엄밀한 수리과학으로 정립하는 장면을 살펴보도록 하자.

---

[7] 이는 RSA 논문에 이론적 토대가 전혀 없다는 뜻은 결코 아니다. 트랩도어 일방향 함수는 이미 비교적 구체적인 정의를 갖고 있던 개념이며, RSA 논문은 그 최초의 설계라는 점에서 매우 중요한 이론적 성취이다. 다만 암호화 스킴의 관점에서 보면, 앞서 언급한 여러 모호함을 고려할 때 RSA 스킴은 안전성 정의 측면에서 아직 완전히 정제되었다고 보기는 어렵다.

참고문헌

1. Rivest, R.L., Shamir, A. and Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), pp.120-126.
2. Rivest, R.L., 2002. ACM A.M. Turing Award Lecture, "The Early Days of RSA: History and Lessons" https://youtu.be/NqqukNkoEVo?si=AeKWrv65U6rwXoHS
3. Rivest, R.L., 2016. ACM A.M. Turing Award Laureate Interviews, "Ron Rivest, 2002 ACM Turing Award Recipient" https://youtu.be/c_j8Y47HoxE?si=w5DpAhXqgW3oMPah
4. Singh, Simon. *The Code Book*. Vol. 7. New York: Doubleday, 1999. (한국에는 출판사 [인사이트]를 통해 <비밀의 언어>라는 제목으로 번역 및 출간되었다.)
5. Levy, Steven. *Crypto: How the code rebels beat the government--saving privacy in the digital age*. Penguin, 2001.
6. Diffie, W., and M. Hellman. "New directions in cryptography." *IEEE Transactions on Information Theory* 22.6 (1976): 644-654.
7. Lenstra, A.K. and Lenstra, H.W., 1993. *The development of the number field sieve* (Vol. 1554). Springer Science & Business Media.
8. Boneh, D. and Venkatesan, R., 1998, May. Breaking RSA may not be equivalent to factoring. In *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 59-71). Berlin, Heidelberg: Springer Berlin Heidelberg.
9. Gardner, M., 1977. A new kind of cipher that would take millions of years to break. *Scientific American*, *237*(8), pp.120-124.
10. Atkins, D., Graff, M., Lenstra, A.K. and Leyland, P.C., 1994, November. The magic words are squeamish ossifrage. In *International Conference on the Theory and Application of Cryptology* (pp. 261-277). Berlin, Heidelberg: Springer Berlin Heidelberg.
11. McHugh, Nathaniel (March 26, 2015). "The Magic Words are Squeamish Ossifrage - factoring RSA-129 using CADO-NFS" https://natmchugh.blogspot.com/2015/03/the-magic-words-are-squeamish-ossifrage.html

이미지 출처

1. https://commons.wikimedia.org/wiki/File:Ronald_L_Rivest_photo.jpg
2. https://commons.wikimedia.org/wiki/File:Adi_Shamir_at_TU_Darmstadt_(2013).jpg
3. https://commons.wikimedia.org/wiki/File:Len-mankin-pic.jpg
4. https://commons.wikimedia.org/wiki/File:RSA_Security_201x_logo.svg
5. https://commons.wikimedia.org/wiki/File:Munitions_T-shirt_(front).jpg