Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.3, Jun. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.3.461

강한 위조 불가능성을 갖는 정수 기반 준동형 메시지 인증 코드*

주 치 홍,† 윤 아 람‡
울산과학기술대학교

A Strongly Unforgeable Homomorphic MAC over Integers*

Chihong Joo,† Aaram Yun‡
Ulsan National University of Science and Technology

요 약

준동형 MAC은 데이터의 무결성을 보호하면서도 제 3자에 의한 데이터 처리를 허용하는 암호학적 개념이다. 본 논문에서는 새로운 준동형 MAC을 제시하고 그 안전성을 증명하도록 한다. 본 논문의 MAC은 정수에 기반하고 있고, 단지 안전한 PRF의 존재성만을 가정하며, 실용적인 Catalano-Fiore 준동형 MAC과 비견할 만한 효율성을 갖는다. 본 방식은 공격자가 MAC 검증 질의를 할 수 있는 환경에서도 위조 불가능하며, 이 안전성은 강한 위조 불가능성을 보이는 방법으로 증명되었다.

ABSTRACT

Homomorphic MAC is a cryptographic primitive which protects authenticity of data, while allowing homomorphic evaluation of such protected data. In this paper, we present a new homomorphic MAC, which is based on integers, relying only on the existence of secure PRFs, and having efficiency comparable to the practical Catalano-Fiore homomorphic MAC. Our scheme is unforgeable even when MAC verification queries are allowed to the adversary, and we achieve this by showing strong unforgeability of our scheme.

Keywords: homomorphic MAC, cloud computing, data authenticity

I. 서 론

최근 각광받고 있는 패러다임인 클라우드 컴퓨팅에서는, 사용자의 데이터 저장 및 정보 처리를 클라우드 환경에 위탁하여 신뢰성과 확장성, 그리고 효율성을 얻어낸다. 하지만 그럼에도 불구하고, 처리해야 할 데이터의 중요성이 큰 경우에는 암호학적 보호가 필요하게 되고, 이는 클라우드 환경에서 제공 가능한 서비스의 범위를 제약하는 요인이 된다. 2009년에 Gentry는 ideal lattice에 기반한 완전 준동형 암호화(fully homomorphic encryption) 방식을 제안하여, 암호화된 데이터도 클라우드 제공자와 같은 제3자에 의한 데이터 처리가 가능함을 입증하였다. 이는 획기적인 방식으로, 향후 충분한 효율성 개선이 이루어진다면 데이터의 기밀성을 유지하면서도 클라우드 컴퓨팅의 장점을 함께 누리는 것을 가능하게 할 것이다. 하지만, 준동형 암호화는 클라우드 상의 데이터의 기밀성(confidentiality)은 보호할 수 있으나, 무결성(integrity)를 보호하지는 못한다.

준동형 메시지 인증 코드(homomorphic MAC, 이하 준동형 MAC으로 표기)는 클라우드 상의 데이

접수일(2014년 3월 20일), 수정일(2014년 5월 26일), 게재확정일(2014년 6월 5일)
* 이 논문은 2011년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(과제 번호: NRF-2011-0025127). 또한 이 논문은 2010년 UNIST(울산과학기술대학교) 연구비에 의하여 연구되었음.
† 주저자, chihongjoo@unist.ac.kr
‡ 교신저자, aaramyun@unist.ac.kr(Corresponding author)

터의 무결성을 보장하면서 동시에 제 3자에 의한 준동형 계산을 가능하게 한다. 하나의 가능한 사용 시나리오는 다음과 같다. 사용자는 자신의 데이터를 특정 길이의 블록 단위로 분할해서(비트 단위도 무방하다) 각각의 블록에 대한 준동형 MAC을 생성한 뒤에, 데이터와 MAC들을 클라우드에 위탁하고, 자신의 컴퓨터에 저장되어 있던 데이터를 삭제한다. 이제 향후 사용자의 데이터의 일부가 필요하거나 혹은 데이터로부터 어떤 함수 f의 계산값이 필요하게 되면, 사용자는 클라우드에게 계산을 요청하게 되고, 클라우드 제공자는 사용자의 데이터로부터 필요한 계산값 v를 구하고, 또한 대응되는 MAC들로부터 준동형성을 이용하여 해당 계산값 v에 대응되는 MAC σ을 유도하여, (v,σ) 순서쌍을 사용자에게 반환한다. 사용자는 MAC 알고리즘을 통해 이 MAC값 σ가 올바른지를 확인하고, 확인이 되는 경우 계산값 v가 올바른 값임을 받아들인다.

Gennaro와 Wichs는 최근 [1]에서 완전 준동형 MAC을 제시하였다. 완전 준동형 암호화 기법을 이용하여 실현한 이들의 방식은 비록 임의의 부울 회로로 표현된 함수를 계산하는 것을 가능하게 하지만, MAC 검증 질의를 허용하지 않는 안전성 모델에 기초하고 있어서, 충분한 안전성을 갖추었다고 보기 어렵다. 또한 방식 자체가 완전 준동형 암호화 기법에 의존하고 있기 때문에, 실용적인 사용과는 아직 거리가 있다.

[2]에서 Catalano와 Fiore는 매우 효율적이고 실용적인 준동형 MAC을 제시하였다. 이들의 방식은 다항식을 이용하여 MAC을 표현하는 것으로, 곱셈을 반복하면 다항식의 크기가 증가하기 때문에 제한된 횟수의 곱셈만이 가능하므로 완전 준동형 방식은 아니지만, 매우 효율적이고, 다른 복잡한 가정에 의존하지 않고 단지 안전한 PRF의 존재성만을 필요로 한다. 또한 이들의 방식은 MAC 검증 질의를 하는 경우에도 안전하다.

본 논문에서는 MAC 검증 질의를 허용하는 안전성 모델을 만족하는, 정수 기반의 MAC을 제시하도록 한다. 본 방식은 Catalano와 Fiore의 방식과 비교하는 것이 가장 적절한데, 그들의 방식이 다항식에 기반하고 있는 것과는 달리 본 방식은 정수 기반으로, 각각의 MAC이 하나의 정수로 표현된다. 역시 곱셈을 반복하면 정수의 크기가 커지므로 제한된 횟수의 곱셈이 가능하다. 하지만 Catalano와 Fiore의 경우와 마찬가지로 PRF의 존재성만을 필요로 하는 매우 효율적인 방식이며, 또한 Catalano와 Fiore의 준동형 MAC의 경우에는 메시지 공간이 소수 p에 대해 잉여계 $\mathbf{Z}_p$로 표현되고, 안전성의 증명을 위해서는 이 소수 p가 충분히 커야 하기 때문에, 메시지 공간을 응용 분야에 따라 마음대로 고르는 것이 불가능하나, 본 방식에서는 작은 메시지 공간을 선택하여도 안전성에 영향을 주지 않는다.

본 논문에서 제시하는 준동형 MAC 역시 검증 질의를 허용하는 환경에서도 안전성을 유지한다. 특히, 본 논문에서는 전통적인 MAC의 '강한 위조 불가능성(strong unforgeability)' 개념을 준동형 MAC으로 확장하여 제시하고, 몇 가지 부가적인 조건이 성립하는 경우, 강한 위조 불가능성을 갖는 준동형 MAC은 검증 질의를 허용하더라도 안전성이 보존된다는 일반적인 결과를 먼저 보인 뒤에, 이를 이용하여 본 논문의 준동형 MAC의 안전성을 증명하는데, 이는 Catalano와 Fiore의 방식과는 또다른 접근 방식으로 향후 준동형 MAC의 안전성 증명에 있어서 한 가지 가능한 전략을 제시한다.

II. 가정 및 정의

2.1 기본적 정의와 가정

본 논문에서 안전성 파라미터(security parameter)는 항상 λ로 표기되고, 다른 파라미터들은 안전성 파라미터의 함수로 주어진다.

본 논문에서 임의의 modulus n에 대해, $x \bmod n$은 $x \equiv y \pmod n$을 만족하는 실구간 $[0,n)$ 상의 유일한 정수 y로 정의된다. 마찬가지로, 잉여계 $\mathbf{Z}_n$ 역시 $[0,n)\cap\mathbf{Z}$와 동일시하도록 한다. 또한, $\lg(x)$는 2를 밑으로 하는 로그함수를 의미한다.

두 개의 확률 변수 X, Y에 대해, 둘 사이의 통계적 거리(statistical distance) $\triangle(X, Y)$는 $\triangle(X, Y) = \frac{1}{2}\sum_z |\Pr[X=z] - \Pr[Y=z]|$로 정의된다.

2.2 준동형 MAC의 정의

여기에서는 준동형 MAC을 정의하고자 한다. 다음의 논의에서는 M, Σ, L, F는 각각 메시지 공간(message space), MAC 공간, 식별자 공간(label space), 그리고 허용 함수 공간(admissible function space)을 가리킨다. 우선 준동형 MAC

의 정의를 위해 식별 프로그램(labeled program)에 대한 정의를 내리도록 한다.

2.2.1 허용 함수 및 식별 프로그램

각각의 준동형 MAC에는 허용함수들의 공간 F가 대응된다. F의 임의의 원소 f는 다항식 시간 내에 계산이 가능한 함수들을 나타내는 구체적인 표현(representation)으로서, 각각의 원소 $f \in F$는 $f : M^l \rightarrow M$의 형태를 갖는다. 여기서 M은 해당 준동형 MAC의 메시지 공간이고, l은 f의 arity라 불리운다.

식별 프로그램 개념은 Gennaro와 Wichs에 의해 [1]에서 처음 제안된 것으로, 준동형 MAC의 정의에 사용된다. 어떤 메시지 $m \in M$에 대한 준동형 MAC을 계산할 때에는 반드시 어떤 '식별자(label)' $\tau \in L$에 대해 계산하도록 되어 있는데, 이때 '식별 프로그램(labeled program)'이란 허용 함수 $f : M^l \rightarrow M$와, 이 f의 각 자리에 어떠한 입력값이 들어가야 하는가에 대한 정보를 한데 묶은 개념이다. 형식적으로는, 식별 프로그램이란 arity가 l인 허용 함수 f와, l개의 식별자 $\tau_1, ..., \tau_l$가 주어졌을 때 이들을 묶은 순서쌍 $P = (f, \tau_1, ..., \tau_l)$ 으로 정의된다.

또한 식별자 $\tau \in L$에 대한 항등 식별 프로그램(Identity labeled program)이란 $I_\tau = (id, \tau)$ 으로 정의되는데, 이때 $id : M \rightarrow M$은 메시지 공간 위에서의 항등함수를 가리킨다.

2.2.2 준동형 MAC

준동형 MAC이란 다음의 네가지 알고리즘들로 이루어진 순서쌍 $\Pi = (Gen, Auth, Eval, Ver)$을 가리킨다.

- $Gen(1^\lambda)$: 안전성 파라미터 λ가 입력으로 주어지면, Gen 알고리즘은 키 쌍 (ek, sk)를 출력한다. 여기서 ek는 준동형 계산을 위한 evaluation key이고, sk는 비밀키이다.
- $Auth(sk, \tau, m)$: 비밀키 sk, 식별자 τ, 그리고 메시지 m이 주어졌을 때, $Auth$ 알고리즘은 메시지 m의 식별자 τ에 대한 MAC 값 σ를 출력한다.
- $Eval(ek, f, \sigma_1, ..., \sigma_l)$: 키 ek, 허용 함수 f, 그리고 f의 arity 개수만큼의 MAC $\sigma_1, ..., \sigma_l$이 주어졌을 때, $Eval$ 알고리즘은 새로운 MAC σ를 출력한다.
- $Ver(sk, (f, \tau_1, ..., \tau_l), m', \sigma')$: 비밀키 sk, 식별 프로그램 $(f, \tau_1, ..., \tau_l)$, 메시지 m', 그리고 MAC σ'이 주어졌을 때, Ver 알고리즘은 0 혹은 1을 출력한다.

위의 알고리즘 중에서, $Eval$과 Ver은 결정적(deterministic) 알고리즘이어야 한다.

위의 포말리즘의 직관적인 의미는 다음과 같다: 사용자는 각각의 메시지 m을 클라우드에 올리기 전에 특정 식별자 τ에 대해 MAC σ를 생성하여 (τ, m, σ) 순서쌍을 클라우드에 전송한다. 여기에서 식별자 τ는 메시지 m값을 지칭하는 고유의 문자열이고, 이것이 필요한 이유는 클라우드에 데이터를 모두 위탁할 것이기 때문에 클라우드에 저장되어 있는 사용자의 데이터를 가리킬 수 있는 방법이 필요하기 때문이다. 그러한 의미에서는 순서쌍 (τ, m, σ)를 클라우드에 전송하는 행위는 데이터 m과 MAC σ를 거대한 해쉬테이블에다가 τ를 키로 하여 저장하는 것에 비교될만 하다: 향후 사용자는 데이터 m과 MAC σ를 더 이상 가지고 있지 않지만, τ를 기억하고 있는 한 클라우드에 저장된 자신의 데이터를 지칭할 수 있고 이에 대해 추가적인 계산을 지시할 수 있다.

여기에서 언급할 사항은, 데이터 m을 저장하기 위해 식별자 τ를 기억해야 한다고 해도 데이터의 위탁이 무의미하지 않다는 점이다: 이는 해쉬테이블에 데이터 m을 저장하기 위해 해쉬키 τ를 기억해야 한다고 해도 이 저장이 무의미하지 않은 것과 마찬가지이다. 많은 경우, 데이터 m의 내용물은 랜덤하지만 식별자 τ는 내부적인 구조를 갖게 구성될 수 있다. 예를 들어, 암호학 강의의 기말고사 성적을 저장하고자 하고, 해당 강좌의 학생 수가 200명이라고 하면, $\tau_1, ..., \tau_{200}$을 정의할 때에 $\tau_1 = crypto \| 1$, $\tau_2 = crypto \| 2$, ..., $\tau_{200} = crypto \| 200$과 같이 정할 수 있고, 이 경우 사용자가 기억할 것은 crypto라는 키워드가 사용되었다는 사실과, 학생 수가 총 200명이라는 것 뿐이다.

위와 같이 사용자의 데이터가 클라우드에 전송된 뒤에는, 사용자가 자신의 데이터에 대해 특정 허용 함수 f를 계산하고 싶으면, 사용자는 적절한 식별 프로그램 $(f, \tau_1, ..., \tau_l)$의 계산을 클라우드에 요청한다. 이 요청의 직관적인 의미는 'l개의 입력을 받는 함수 f의 계산 결과값을 돌려주되, 그 i번째 입력은 식별자 τ_i

에 대하여 MAC 값이 계산된 적이 있는 나의 데이터 m_i로 하라'가 된다. 즉, 이전에 사용자가 $Auth(sk,\tau_i,m_i)$를 계산하여 그 결과를 클라우드에 보낸 적이 있다고 하면, 이 식별 프로그램의 올바른 계산값은 $f(m_1,...,m_l)$이 된다. 클라우드 제공자는 이 요청을 받은 뒤, 우선 함수값 $m'=f(m_1,...,m_l)$을 계산하고, 또한 기존에 저장해 둔 m_i의 MAC 값 σ_i를 $Eval$ 알고리즘에 적용하여 함수값 m'에 대응하는 MAC 값 σ'을 계산하여, 순서쌍 (m',σ')을 사용자에게 반환한다. 여기에서 σ'는 반환된 값 m'이 올바른 함수값 $f(m_1,...,m_l)$이라는 증명이 된다. 사용자는 이제 Ver 알고리즘을 이용하여 이 증명이 올바른지를 확인하고, 결과값이 1인 경우 그 증명을 받아들여 $m'=f(m_1,...,m_l)$임을 신뢰하게 된다.

준동형 MAC이 올바르기 위해 만족해야 할 성질은 다음과 같다.

- $Ver(sk,I_\tau,m,Auth(sk,\tau,m))=1$이 임의의 τ, m에 대해 성립: 즉, 적법하게 생성한 MAC은 검증을 통과
- 임의의 f, $\tau_1,...,\tau_l$, $m_1,...,m_l$에 대해, $\sigma_i\leftarrow Auth(sk,\tau_i,m_i)$로 MAC을 계산 후, 다음으로 $\sigma'\leftarrow Eval(ek,f,\sigma_1,...,\sigma_l)$를 구하면, $Ver(sk,(f,\tau_1,...,\tau_l),f(m_1,\cdots,m_l),\sigma')=1$이 성립: 적법하게 준동형 계산된 MAC은 검증을 통과

또한 준동형 MAC은 간결성(succinctness)을 만족해야 한다. 이는 MAC의 크기가 허용 함수 f에 의존하지 말고 단지 안전성 파라미터의 어떤 다항식 이하여야 한다는 의미이다. 이는, 여태까지의 정의를 모두 만족함에도 불구하고 무의미한 준동형 MAC을 허용하지 않기 위해서이다.

2.3 상수성 검사 가능성(constant testability)

준동형 MAC Π와 그 evaluation key ek를 하나 고정하자. Arity가 l인 허용 함수 f와, 집합 $\{1,\cdots,l\}$의 부분집합 I, 그리고 특정 메시지들의 순서쌍 $(m_i)_{i\in I}$, 그에 대응되는 MAC들의 순서쌍 $(\sigma_i)_{i\in I}$가 주어졌을 때에, 다음을 정의하도록 하자:

- 메시지 $(m_i)_{i\in I}$들이 주어질 때, 이들에 대한 허용 함수 f의 부분 적용(partial application)이란, $f'((m_j)_{j\notin I}):=f(m_1,\cdots,m_l)$로 정의되는 함수 $f':M^{l-|I|}\rightarrow M$을 의미하고, 이를 $App(f,(m_i)_{i\in I})$로 표기한다.
- MAC $(\sigma_i)_{i\in I}$들이 주어질 때, 이들에 대한 허용 함수 f의 부분 준동형 계산(partial homomorphic evaluation)이란, $e((\sigma_j)_{j\notin I}):=Eval(ek,f,\sigma_1,\cdots,\sigma_l)$로 정의되는 함수 $e:\Sigma^{l-|I|}\rightarrow\Sigma$를 의미하고, 이를 $Eval(f,(\sigma_i)_{i\in I})$로 표기한다.

즉, $App(f,(m_i)_{i\in I})$는 허용 함수 f에다가 $i\in I$들마다 입력 m_i를 미리 넣어둔 것으로, $App(f,(m_i)_{i\in I})$는 이제 남아있는 입력 $(m_j)_{j\notin I}$들의 함수가 된다. 마찬가지로, $Eval(f,(\sigma_i)_{i\in I})$는 허용 함수 f의 준동형 MAC 계산 $Eval(ek,f,\sigma_1,\cdots,\sigma_l)$의 일부 자리들에 입력 σ_i를 미리 넣어둔 것으로, $Eval(f,(\sigma_i)_{i\in I})$는 이제 남아있는 입력들의 함수가 된다. 특히, $I=\{1,\cdots,l\}$인 경우 $Eval(f,(\sigma_i)_{i\in I})$는 상수함수임이 분명하다.

어떤 준동형 MAC Π가 상수성 검사 가능성(constant testability, 이하 CT로 표기)를 만족한다는 것은, 위와 같은 $Eval(f,(\sigma_i)_{i\in I})$, 그리고 $App(f,(m_i)_{i\in I})$가 상수함수인지 아닌지의 여부를 검사하는 효율적인 확률적 알고리즘이 존재한다는 것을 의미한다. 보다 정확히 말하자면, 어떤 PPT 알고리즘 $DEval$, $DApp$이 존재하여,

$DEval(ek,(f,\tau_1,\cdots,\tau_l),I,(\sigma_i)_{i\in I})=1$인데도 불구하고 $Eval(f,(\sigma_i)_{i\in I})$가 상수함수가 아니거나, 혹은 $DEval(ek,(f,\tau_1,\cdots,\tau_l),I,(\sigma_i)_{i\in I})=0$인데도 불구하고 $Eval(f,(\sigma_i)_{i\in I})$가 상수함수일 확률이 무시할만하고, 또한 $DApp((f,\tau_1,\cdots,\tau_l),I,(m_i)_{i\in I})=1$이면서 $App(f,(m_i)_{i\in I})$가 상수함수가 아니거나, 혹은 $DApp((f,\tau_1,\cdots,\tau_l),I,(m_i)_{i\in I})=0$인데도 불구하고 $App(f,(m_i)_{i\in I})$가 상수함수일 확률도 무시할 만할 때, Π는 상수성 검사 가능성을 만족한다고 한다.

다른 말로, 효율적으로 $Eval(f,(\sigma_i)_{i\in I})$과 $App(f,(m_i)_{i\in I})$의 상수성을 판정할 수 있는 알고리즘들이 있고, 이들의 오류 확률이 무시할만 할 때 준동형 MAC Π가 CT 성질을 만족한다고 정의한다.

경우에 따라서는, $App(f,(m_i)_{i\in I})$의 상수성을 판정하기는 어려우나 $Eval(f,(\sigma_i)_{i\in I})$의 상수성을 판정

하기는 어렵지 않은 경우가 있다. 따라서, 위와 같은 $Eval(f,(\sigma_i)_{i\in I})$의 상수성을 효율적으로 판정할 수 있는 경우, 준동형 MAC Π는 MAC에 대한 상수성 검사 가능성(MCT)을 만족한다고 정의한다.

2.4 준동형 MAC의 안전성 정의

우리가 사용할 준동형 MAC의 기본적인 안전성 정의는 Catalano와 Fiore에 의해 [2]에서 제시된 준동형 MAC의 위조 불가능성(unforgeability)이다. 준동형 MAC Π와 공격자 A에 대해, 다음의 게임 $UF^{\Pi,A}(1^\lambda)$을 생각하자:

1. 준비: 키 쌍 $(ek,sk)\leftarrow Gen(1^\lambda)$을 생성한 뒤 evaluation key ek가 공격자 A에게 주어진다. 집합 S가 공집합으로 초기화된다. 이 S는 공격자의 질의와 그에 대한 답변의 기록을 저장하는 집합이다.
2. 질의: 공격자 A는 임의의 식별자-메시지 쌍 (τ,m)에 대해 원하는 횟수만큼 MAC 생성 질의를 할 수 있다. 만일 $(\tau,m,\sigma)\in S$를 만족하는 어떤 σ가 존재하면, 이 질의는 무시된다. 만일 그렇지 않다면, $\sigma\leftarrow Auth(sk,\tau,m)$이 계산되고, S는 $S\leftarrow S\cup\{(\tau,m,\sigma)\}$로 갱신되고, 공격자에게는 질의의 답으로 MAC σ가 주어진다.
3. 위조 시도: 최종적으로, 공격자 A는 어떤 위조 시도 $((f,\tau_1,...,\tau_l),m',\sigma')$를 출력한다. 만일 다음의 조건들이 만족되면 이 게임은 최종적으로 1을 출력하고, 그렇지 않은 경우에는 이 게임은 최종적으로 0을 출력하게 된다: 우선, $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=1$가 성립해야 하고, 그 외에도 다음의 두 조건 중 하나가 만족되어야 한다.
   - 제 1형 위조: 각각의 $i=1,\cdots,l$ 중에서 $(\tau_i,m_i,\sigma_i)\in S$를 만족하는 m_i,σ_i가 존재하는 i들의 집합을 I라고 할 때, 허용 함수 f의 부분 적용 $App(f,(m_i)_{i\in I})$가 상수함수가 아니다. 혹은,
   - 제 2형 위조: 위에서 $App(f,(m_i)_{i\in I})$가 함수값 m^*를 갖는 상수함수이지만, $m^*\neq m'$이다.

준동형 MAC Π가 위조불가능(unforgeable)하다는 것은, 임의의 PPT 공격자 A에 대해 그 advantage $Adv^{UF}_{\Pi,A}(\lambda)=\Pr[UF^{\Pi,A}(1^\lambda)=1]$가 안전성 파라미터 λ에 대해 무시할 만한(negligible) 함수인 경우를 가리킨다. 이 논문에서는 이 안전성을 UF 안전성이라 명명한다.

위의 안전성 정의는 다음과 같이 두 가지 측면에서 확장될 수 있다. 일단 검증 질의를 허용하는 위조 불가능성(unforgeability with verification queries)은 게임 $UFV^{\Pi,A}(1^\lambda)$에 의해 비슷한 방법으로 정의되는데, 게임 $UFV^{\Pi,A}(1^\lambda)$은 게임 $UF^{\Pi,A}(1^\lambda)$와 거의 동일하지만, 차이가 있다면 공격자 A가 MAC 생성 질의 뿐만이 아니라 임의 횟수의 검증 질의 $((f,\tau_1,...,\tau_l),m',\sigma')$를 하고 그 결과값 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')$를 돌려받을 수 있다는 것이다. 이 안전성은 편의상 UFV 안전성이라고 정의하자. 역시 이 경우에도 공격자의 advantage를 $Adv^{UFV}_{\Pi,A}(\lambda)=\Pr[UFV^{\Pi,A}(1^\lambda)=1]$로 정의할 수 있다.

일반적인 MAC이나 전자 서명에서 강한 위조불가능성 개념이 정의되어 있는 것에 착안하여, 다음과 같이 준동형 MAC에 대해서도 강한 위조불가능성 개념을 정의하는 것이 가능하다. 이는 게임 $SUF^{\Pi,A}(1^\lambda)$에 의해 비슷한 방법으로 정의되는데, 이 게임은 게임 $UF^{\Pi,A}(1^\lambda)$와 완전히 동일하지만 마지막 단계에서 공격자의 위조 시도를 판정할 때에 다음과 같은 변형된 논리를 사용한다:

3. 위조 시도: 최종적으로, 공격자 A는 어떤 위조 시도 $((f,\tau_1,...,\tau_l),m',\sigma')$를 출력한다. 만일 다음의 조건이 만족되면 이 게임의 최종 출력은 1이 되고, 그렇지 않은 경우 이 게임은 최종적으로 0을 출력하게 된다: 우선 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=1$가 성립해야 하고, 그 외에도 다음의 조건들 중 하나가 만족되어야 한다.
   - 제 1형 강한 위조(strong forgery): 각각의 $i=1,\cdots,l$ 중에서 $(\tau_i,m_i,\sigma_i)\in S$를 만족하는 m_i,σ_i가 존재하는 i들의 집합을 I라고 할 때, 허용 함수 f의 부분 준동형 계산 $Eval(f,(\sigma_i)_{i\in I})$가 상수함수가 아니다. 혹은,
   - 제 2형 강한 위조: $Eval(f,(\sigma_i)_{i\in I})$가 함수값 σ^*를 갖는 상수함수이나, $\sigma^*\neq\sigma'$이다. 혹은,
   - 위조: 위조 시도 $((f,\tau_1,...,\tau_l),m',\sigma')$가 성공적인 위조이다.

준동형 MAC Π가 강한 위조불가능(strongly

unforgeable)하다는 것은, 임의의 PPT 공격자 A에 대해 다음과 같이 정의되는 그 이점(advantage) $Adv_{\Pi,A}^{SUF}(\lambda)=\Pr[SUF^{\Pi,A}(1^\lambda)=1]$가 안전성 파라미터 λ에 대해 무시할 만한 함수인 경우를 가리킨다. 이 논문에서는 이 안전성을 SUF 안전성이라 명명한다.

마지막으로, 위의 두 가지 확장을 모두 적용한 안전성 SUFV, 즉 검증 질의를 허용하는 강한 위조불가능성을 정의할 수 있다. 이 안전성은 게임 $SUFV^{\Pi,A}(1^\lambda)$에 의해 정의되는데, 이 게임은 $SUF^{\Pi,A}(1^\lambda)$와 거의 동일하지만 임의의 검증 질의를 허용한다는 점이 다르다. 역시 이 안전성 모델에서도 공격자가 갖는 이득(advantage)를 $Adv_{\Pi,A}^{SUFV}(\lambda)=\Pr[SUFV^{\Pi,A}(1^\lambda)=1]$로 정의할 수 있다.

일반적으로 MAC의 표준적인 안전성 정의는 강한 위조불가능성보다는 위조불가능성으로 주어진다. 하지만 검증 오라클은 비교적 현실적인 공격 시나리오이므로, UF보다는 UFV가 보다 더 적절한 안전성 정의라고 간주될 수 있다.

2.5 안전성 개념들 간의 관계

여기에서는 앞에서 정의한 다양한 안전성 개념들 사이의 관계에 관해 논하고자 한다. 우선, 검증 질의를 허용하는 안전성은 자명하게 검증 질의를 허용하지 않는 안전성을 함의한다:

정리 1. 어떤 준동형 MAC Π가 UFV 안전성을 만족하면, Π는 UF 안전성을 만족한다. 또한, Π가 SUFV 안전성을 만족하면, Π는 SUF 안전성을 만족한다.

증명. 자명하다. □

또한, 강한 위조불가능성은 (약한) 위조불가능성을 함의한다. 이 또한 정의 자체에 의해 분명하다.

정리 2. 어떤 준동형 MAC Π가 SUF 안전성을 만족하면, Π는 UF 안전성을 만족한다. 또한, Π가 SUFV 안전성을 만족하면, Π는 UFV 안전성을 만족한다.

증명. SUF 안전성의 정의에서, 성공적인 위조는 성공적인 강한 위조의 특별한 경우이므로, 강한 위조를 만들기 어려우면 위조 또한 만들기 어렵다. □

이제, SUF를 만족하는 준동형 MAC Π는 적절한 조건 하에서 SUFV 안전성 또한 만족한다는 다음의 정리를 보이도록 한다:

정리 3. 어떤 준동형 MAC Π가 SUF 안전성을 만족하고 또한 상수성 검사 가능성(CT)을 만족하면, Π는 또한 SUFV 안전성도 만족한다.

증명. Π가 상수성 검사 가능성을 만족하므로, 허용 함수의 부분 적용 혹은 준동형 계산의 상수성을 무시할 만한 오류 확률을 가지고 판정하는 효율적인 알고리즘이 존재한다. 게임 전체를 통틀어 이 상수성 검사 과정에서 오류가 일어날 확률 또한 무시할만 하므로, 일반성을 잃지 않고 상수성 검사가 오류 없이 일어난다고 가정하자.

이제, A가 $SUFV=SUFV^{\Pi,A}(1^\lambda)$ 게임에 참여하는 임의의 PPT 공격자라고 하자. 일반성을 잃지 않고, A가 정확히 $q=q(\lambda)$ 회의 검증 질의를 한다고 가정할 수 있다. 각각의 $k\in\{0,1,\cdots,q\}$에 대해, $SUFV_k$를 $SUFV$와 다른 면에서는 동일하지만, 처음의 k회의 검증 질의에 대한 답변이 다음과 같은 방식으로 주어진다는 점만 다른 게임으로 정의하도록 하자:

검증 시뮬레이션: 공격자 A의 검증 질의를 $((f,\tau_1,...,\tau_l),m',\sigma')$라 할 때, 각각의 $i=1,\cdots,l$ 중에서 $(\tau_i,m_i,\sigma_i)\in S$를 만족하는 m_i,σ_i가 존재하는 i들의 집합을 I라고 하면, 만일 $Eval(f,(\sigma_i)_{i\in I})$가 σ'를 값으로 갖는 상수함수이고, 또한 $App(f,(m_i)_{i\in I})$가 m'를 값으로 갖는 상수함수이면, 1을 검증 질의의 답으로 반환하고, 그렇지 않으면 0을 반환한다.

정의에 의하면, $SUFV_0$은 $SUFV$와 동일하고, 따라서 두 게임에서 공격자 A가 강한 위조에 성공할 확률은 동일하다. 즉, $Adv_{\Pi,A}^{SUFV_0}(\lambda)=Adv_{\Pi,A}^{SUFV}(\lambda)$이 성립한다.

한편, $SUFV_q$에서 공격자의 검증 질의는 일체 위

의 검증 시뮬레이션에 의해 답변된다. 이 검증 시뮬레이션은 어떤 비밀 정보도 가지고 있지 않고 효율적으로 계산 가능하므로, $SUFV_q$ 게임에서 공격자 A는 검증 질의로부터 어떠한 유용한 추가적인 정보도 얻어내지 못한다. 이를 형식적으로 표현하면, 우리는 공격자 A로부터 $Adv_{\Pi,A}^{SUFV_0}(\lambda)=Adv_{\Pi,B}^{SUF}(\lambda)$를 만족하는 PPT 공격자 B를 구성할 수 있다.

각각의 $k=1,\cdots,q$에 대해, $Adv_{\Pi,A}^{SUFV_{k-1}}(\lambda)$과 $Adv_{\Pi,A}^{SUFV_k}(\lambda)$의 차는, 공격자 A의 k번째 검증 질의에 대한 검증 시뮬레이션이 실패할 확률, 즉, 검증 시뮬레이션의 결과가 실제 검증 오라클의 결과와 다를 확률에 의해 bound된다. 강한 위조 불가능성의 정의로부터, 검증 시뮬레이션이 실패하는 경우는 정확히 공격자 A의 질의가 강한 위조인 경우임을 알 수 있다. 우선, 검증 시뮬레이션이 실패한다는 것은 다음의 둘 중 하나가 성립할 때이다.

1) $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=0$이지만 $Eval(f,(\sigma_i)_{i\in I})$가 σ'를 값으로 갖는 상수함수이고, 또한 $App(f,(m_i)_{i\in I})$가 m'를 값으로 갖는 상수함수이거나, 아니면

2) $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=1$이지만 $Eval(f,(\sigma_i)_{i\in I})$가 상수함수가 아니거나 상수함수임에도 σ'가 아닌 상수값을 갖거나, 혹은 $App(f,(m_i)_{i\in I})$가 상수함수가 아니거나 상수함수임에도 m'가 아닌 상수값을 갖거나

하지만, 1)의 경우에는 준동형 MAC의 기본 성질에 의해 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=0$일 수 없으므로, 1)의 경우는 발생하지 않는다. 즉, 검증 시뮬레이션이 실패한다는 것은 정확히 2)의 경우인데, 이는 강한 위조의 정의와 동일하다.

따라서, 공격자 A의 k번째 검증 질의를 이용하여 SUF 안전성을 공격하는 공격자 C를 구성하는 것이 가능하다. 구체적으로, 공격자 C는 공격자 A가 k번째 검증 질의를 하기 직전까지 A를 동작시키고, 이전까지의 검증 질의를 위에 묘사한 검증 시뮬레이션에 의해 처리한다. 공격자 A가 k번째 검증 질의 $((f,\tau_1,...,\tau_l),m',\sigma')$를 하면, 공격자 C는 A의 실행을 중단시키고, $((f,\tau_1,...,\tau_l),m',\sigma')$를 자기 자신의 위조 시도로 출력한다. 그렇다면, $\left|Adv_{\Pi,A}^{SUFV_{k-1}}(\lambda)-Adv_{\Pi,A}^{SUFV_k}(\lambda)\right|\le Adv_{\Pi,C}^{SUF}(\lambda)$이 성립하고, 준동형 MAC Π가 SUF 안전성을 만족하기 때문에 이는 무시할 만한 값이 된다.

여태까지를 정리하면,

$$\begin{aligned}Adv_{\Pi,A}^{SUFV}(\lambda) &\le Adv_{\Pi,B}^{SUF}(\lambda)+\left|Adv_{\Pi,A}^{SUFV}(\lambda)-Adv_{\Pi,B}^{SUF}(\lambda)\right|\\ &=negl(\lambda)+\left|Adv_{\Pi,A}^{SUFV_0}(\lambda)-Adv_{\Pi,A}^{SUFV_q}(\lambda)\right|\\ &=negl(\lambda)+\sum_{k=1}^{q}\left|Adv_{\Pi,A}^{SUFV_k}(\lambda)-Adv_{\Pi,A}^{SUFV_{k-1}}(\lambda)\right|\\ &=negl(\lambda)+q\cdot negl(\lambda)=negl(\lambda)\end{aligned}$$

위에서, $negl(\lambda)$은 안전성 파라미터 λ에 대한 어떤 무시할 만한 함수를 가리킨다.

따라서 $Adv_{\Pi,A}^{SUFV}(\lambda)$ 역시 임의의 PPT 공격자 A에 대해 무시할 만하고, 따라서 준동형 MAC Π는 SUFV를 만족한다. □

다음 장에서 우리가 제시하는 준동형 MAC의 경우에는 불행히도 완전한 상수성 검사 가능성을 만족하지 못하지만, MAC에 대한 상수성 검사는 가능하다. 따라서 이러한 경우에 대한 결과 또한 제시하고자 한다. 우선 간단한 정의와 보조정리를 제시한다.

정의 4. 어떤 준동형 MAC이 메시지 유일성(message uniqueness)을 갖는다는 것은, 임의의 sk, $(f,\tau_1,...,\tau_l)$, m, m', σ에 대해, 만일 $Ver(sk,(f,\tau_1,...,\tau_l),m,\sigma)=1$이 성립할 뿐 아니라 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma)=1$이 동시에 성립한다면 반드시 $m=m'$이어야 함을 의미한다.

보조정리 5. 어떤 준동형 MAC Π이 메시지 유일성을 만족한다고 가정하자. Arity가 l인 허용 함수 f와, 집합 $\{1,\cdots,l\}$의 부분집합 I, 그리고 특정 메시지들의 순서쌍 $(m_i)_{i\in I}$, 그에 대응되는 MAC들의 순서쌍 $(\sigma_i)_{i\in I}$가 주어졌을 때에, 만일 $Eval(f,(\sigma_i)_{i\in I})$가 상수함수이면 $App(f,(m_i)_{i\in I})$ 역시 상수함수가 된다.

증명. 만일 그렇지 않다고 가정하자.
$f'=App(f,(m_i)_{i\in I})$라고 놓으면, 어떤 $(m_j)_{j\notin I}$와 $(m'_j)_{j\notin I}$가 있어서 $f'(m_j)_{j\notin I}\neq f'(m'_j)_{j\notin I}$을 만족

한다. $m = f'(m_j)_{j \notin I}$, $m' = f'(m'_j)_{j \notin I}$라고 놓자. 각각의 $j \notin I$에 대해 $\sigma_j \leftarrow Auth(sk, \tau_j, m_j)$, $\sigma'_j \leftarrow Auth(sk, \tau_j, m'_j)$로 계산하자. 이제, 상수함수 $Eval(f, (\sigma_i)_{i \in I})$의 함수값을 σ라고 하면, 준동형 MAC Π가 올바르기 때문에 $Ver(sk, (f, \tau_1, ..., \tau_l), m, \sigma) = 1$이 성립할 뿐만 아니라 $Ver(sk, (f, \tau_1, ..., \tau_l), m', \sigma) = 1$ 역시 성립해야 하고, 그렇다면 메시지 유일성에 의해 $m = m'$, 즉 $f'(m_j)_{j \notin I} = f'(m'_j)_{j \notin I}$이 성립해야 하고, 이는 모순이다. 따라서 $App(f, (m_i)_{i \in I})$ 또한 상수함수일 수밖에 없다. □

정리 6. 어떤 준동형 MAC Π가 SUF 안전성, MAC에 대한 상수성 검사 가능성(MCT), 그리고 메시지 유일성을 만족하면, Π는 또한 SUFV 안전성도 만족한다.

증명. 이 정리의 증명은 정리 3과 유사하다. 정리 3의 핵심은 SUF 안전성을 만족하는 준동형 MAC Π가 주어졌을 때에, 이에 대한 MAC 검증 질의를 무시할만한 확률만큼의 예외를 제외하고는 올바르게 처리할 수 있는 검증 시뮬레이션 알고리즘을 제시하는 것이었다. 정리 6의 경우에는 온전한 상수성 검사 가능성을 만족하는 것이 아니라 단지 MAC에 대한 상수성 검사만이 가능하기 때문에 정리 3의 검증 시뮬레이션 알고리즘을 그대로 사용할 수는 없지만, 보조정리 5의 결과를 이용하면 유사한 검증 시뮬레이션을 다음과 같이 구성할 수 있다.

검증 시뮬레이션: 공격자 A의 검증 질의를 $((f, \tau_1, ..., \tau_l), m', \sigma')$라 할 때, 각각의 $i = 1, \cdots, l$ 중에서 $(\tau_i, m_i, \sigma_i) \in S$를 만족하는 m_i, σ_i가 존재하는 i들의 집합을 I라고 하면, 만일 $Eval(f, (\sigma_i)_{i \in I})$가 상수함수가 아니면 0을 검증 질의의 답으로 반환한다. 만일 그렇지 않고 $Eval(f, (\sigma_i)_{i \in I})$가 σ'를 값으로 갖는 상수함수라면, 보조정리 1에 의하면 $f' = App(f, (m_i)_{i \in I})$ 역시 상수함수가 된다. 임의로 $(m_j)_{j \notin I}$를 선택하여 $f'(m_j)_{j \notin I}$을 계산함으로 상수함수 $App(f, (m_i)_{i \in I})$의 함수값을 확인해 보아서, 만일 그 함수값이 m'이면, 1을 반환하고, 그렇지 않은 경우에는 0을 반환한다.

이 검증 시뮬레이션은 정리 3의 검증 시뮬레이션과 동일하게, $Eval(f, (\sigma_i)_{i \in I})$가 σ'를 값으로 갖는 상수함수이고 $App(f, (m_i)_{i \in I})$가 m'를 값으로 갖는 상수함수이면 1을 출력하고, 그렇지 않다면 0을 출력하지만, $App(f, (m_i)_{i \in I})$가 상수함수인지를 보조정리 1에 의존하여 간접적으로 확인한다. 결과적으로 이 검증 시뮬레이션 또한 효율적으로 계산 가능하며, 정리 3의 경우와 동일한 증명을 이 검증 시뮬레이션을 통해 구성할 수 있다. □

앞에서 밝힌 대로, 준동형 MAC의 가장 적절한 안전성 정의는 UFV로 간주될 수 있다. 정리 3과 4가 말하는 것은, 만일 준동형 MAC Π가 SUF 안전성을 만족한다면, 이 Π가 추가적으로 CT 성질을 만족하거나, 아니면 MCT와 메시지 유일성을 만족하기만 하면, Π는 자동으로 SUFV 안전성도 만족하게 된다는 것이고, 이러한 경우 정리 2에 의해 UFV 안전성 또한 만족되게 된다. 따라서, 어떠한 준동형 MAC의 안전성을 증명하는 한 가지 매력적인 전략은, 해당 MAC의 SUF 안전성을 증명하는 것임을 알 수 있다.

III. 제안하는 기법

본 장에서는 SUF 안전성을 만족하는 준동형 MAC을 제안하고자 한다. 본 준동형 MAC은 approximate GCD 가정 등의 복잡한 가정에 의존하지 않고, 단지 의사난수함수(pseudorandom function, 이하 PRF)의 존재성에만 의존한다.

3.1 허용 함수의 묘사

아래에 묘사할 준동형 MAC Π의 허용 함수는 산술 회로(arithmetic circuit)들로 주어지고, 이러한 회로는 다음의 두 게이트들을 이용하여 구성된다:

- 덧셈 게이트: 두 개의 산술 회로 C, C'이 주어지면, 덧셈 게이트를 통해 이들의 출력값을 더한 회로 $C + C'$을 구성할 수 있다.
- 곱셈 게이트: 두 개의 산술 회로 C, C'이 주어지면, 곱셈 게이트를 통해 이들의 출력값을 곱한 회로 $C \times C'$을 구성할 수 있다.

또한, 다음의 기본적인 산술 회로들이 주어진다:

- 상수 회로: 임의의 $a \in \{0,1,\cdots,N-1\}$에 대해,

$C_{(a)}$는 입력이 없고 a를 출력으로 갖는 산술 회로이다.

- 항등 회로: w가 임의의 입력선(input wire)이면, 항등 회로 C_w는 입력선 w로 들어온 임의의 입력값 x를 그대로 다시 출력하는 항등함수를 나타내는 산술 회로이다.

위에서 파라미터 N은 메시지 공간 $\mathbf{Z}_N$를 기술한다. 이후에 묘사되겠지만, N은 다른 어떤 파라미터 η에 대해 $N \leq 2^{\eta-1}$을 만족하도록 선택된다.

임의의 산술 회로 C는 자연스럽게 입력에 대한 다변수 정수 계수 다항식으로 간주될 수 있고, 따라서 차수 $\deg(C)$를 생각할 수 있다. 임의의 산술 회로 C는, 입력이 $\mathbf{Z}_N$ 상의 메시지들로 주어졌을 때 modulo N 계산을 통해 자연스럽게 메시지 공간 $\mathbf{Z}_N$ 위의 함수로 간주될 수 있다. 또한, 우리가 묘사할 준동형 MAC Π의 MAC들은 하나의 정수로 묘사되는데, 임의의 산술 회로 C는 다변수 정수 계수 다항식이므로, MAC들을 입력으로 삼아 계산이 가능하다.

각각의 산술 회로 C에 다음과 같은 방법으로 자연수 값 $bd(C)$를 대응시키자:

- $bd(C_w) := \rho + \eta$
- $bd(C_{(a)}) := \eta$
- $bd(C \times C') := bd(C) + bd(C')$
- $bd(C + C') := 1 + \max\{bd(C), bd(C')\}$

여태까지의 정의에 의하면, 산술 회로 $C(x_1, \cdots, x_l)$의 각 입력 x_i가 $0 \leq x_i \leq 2^{\rho}N$을 만족하면, $0 \leq C(x_1, \cdots, x_l) \leq 2^{bd(C)}$가 성립함을 쉽게 귀납법에 의해 확인할 수 있다. 또한, 역시 간단한 귀납법에 의해, 임의의 산술 회로에 대해 $\deg(C) \leq bd(C)$가 성립함을 확인할 수 있다.

3.2 준동형 MAC의 묘사

이하에서 λ는 안전성 파라미터이고, 본 방식에 사용되는 모든 파라미터는 λ로부터 효율적으로 계산 가능한 함수로 주어진 것으로 간주하자. 본 준동형 MAC에 사용되는 파라미터로는 λ, ρ, η, β, N이 있다.

ρ, η는 알고리즘에 사용된 난수들을 선택하기 위해 사용되는 파라미터이고, β는 허용 함수를 묘사하기 위한 파라미터이다. N은 메시지 공간 $\mathbf{Z}_N$을 기술하기 위한 파라미터이다. 이 준동형 MAC에서 사용되는 PRF $F(k,\tau)$에서 PRF 키 k와 메시지 τ의 길이는 λ 비트이고, PRF의 값 $F(k,\tau)$의 길이는 ρ 비트라고 가정하자. 또한 이 경우 자연스럽게 $F(k,\tau)$가 $\{0,1,\cdots,2^{\rho}-1\}$에서 값을 갖는다고 간주할 수 있다.

준동형 MAC Π의 공간들은 다음과 같다.

- 메시지 공간 M은 $\mathbf{Z}_N$으로 주어진다.
- MAC 공간 Σ는 $0 \leq \sigma \leq 2^{\beta}$를 만족하는 모든 정수 σ들의 집합으로 주어진다.
- 식별자 공간 L은 $\{0,1\}^{\lambda}$로 주어진다.
- 허용 함수 공간 F는 $bd(f) \leq \beta$를 만족하는 모든 산술 회로 f들의 집합으로 주어진다.

이제 준동형 MAC Π의 알고리즘들을 기술하도록 한다.

- $Gen(1^{\lambda})$: 안전성 파라미터 λ가 입력으로 주어지면, PRF의 비밀키 k를 $\{0,1\}^{\lambda}$에서 랜덤하게 고르고, 랜덤한 η 비트 소수 p를 고른 뒤 $(p \in (2^{\eta-1}, 2^{\eta}) \cap \mathrm{PRIME})$, evaluation key $ek = \lambda$, 비밀키 $sk = (k,p)$를 계산하고, 키 쌍 (ek, sk)를 출력한다.
- $Auth(sk, \tau, m)$: 비밀키 $sk = (k,p)$, 식별자 τ, 그리고 메시지 m이 주어졌을 때, 먼저 PRF의 함수값 $r = F(k,\tau)$를 계산하고, $a = N^{-1}(r-m) \bmod p$를 계산한 뒤, 난수 q를 구간 $[0, \lfloor 2^{\rho}/p \rfloor) \cap \mathbf{Z}$에서 균일 분포로 고른 뒤 $\sigma = (pq+a)N + m$을 출력한다.
- $Eval(ek, f, \sigma_1, ..., \sigma_l)$: $f(\sigma_1, \cdots, \sigma_l)$을 출력한다.
- $Ver(sk, (f, \tau_1, ..., \tau_l), m', \sigma')$: 비밀키 $sk = (k,p)$, 식별 프로그램 $(f, \tau_1, ..., \tau_l)$, 메시지 m', 그리고 MAC σ'이 주어졌을 때, 우선 각각의 $i = 1, \cdots, l$에 대해 $r_i = F(k, \tau_i)$를 계산하고, $r = f(r_1, \cdots, r_l)$을 계산한 뒤, 만일 $\sigma' \equiv m' \pmod N$이고 또한 $\sigma' \equiv r \pmod p$이면 1을 출력하고, 그렇지 않으면 0을 출력한다.

3.3 파라미터의 선택

이하에서 λ는 안전성 파라미터이고, 본 방식에 사용되는 모든 파라미터는 λ로부터 효율적으로 계산 가능하며, 다음을 만족해야 한다.

- η는 $\eta=\omega(\lg\lambda)$를 만족해야 하고, 이는 정리 13의 증명을 위해 필요하다.
- ρ는 $\rho=\eta+\omega(\lg\lambda)$를 만족해야 하고, 이는 역시 정리 13의 증명을 위해 필요하다.
- β는 $\beta\geq\rho+\eta$를 만족하는 다항식이어야 하고, 이는 상수 회로와 항등 회로를 허용하기 위해서이다. β는 허용 함수 공간 F의 크기를 결정하므로, 특정 응용이 요구하는 만큼 충분히 클 필요가 있다.
- N은 $N\leq 2^{\eta-1}$을 만족해야 한다. 이 경우 η 비트 소수 p에 대해 $\gcd(p,N)=1$이 성립된다. N의 선택이 메시지 공간 $\mathbf{Z}_N$을 결정한다.

이상의 파라미터들 중, 그리스 문자로 주어진 λ, η, ρ, β는 안전성 파라미터 λ에 대한 어떤 다항식 이하여야 하고, N은 λ에 대한 계산 가능한 함수이기만 하면 된다.

위의 조건을 모두 만족하는 파라미터 선택의 한 가지 예로는, $\eta=\lambda$, $\rho=2\lambda$, $\beta\geq 3\lambda$ 등이 있다.

3.4 올바름(Correctness)

정리 7. 이상의 준동형 MAC Π는 간결하다.

증명. MAC 계산식 $\sigma=(pq+a)N+m$에서, $pq+a\leq p\left(\left\lfloor\frac{2^\rho}{p}\right\rfloor-1\right)+(p-1)\leq p\left\lfloor\frac{2^\rho}{p}\right\rfloor-1$이고, 따라서 $\sigma\leq\left(p\left\lfloor\frac{2^\rho}{p}\right\rfloor-1\right)N+(N-1)<2^\rho N$이 성립한다. 따라서, $Auth$알고리즘에 의해 출력되는 임의의 MAC이 $0\leq\sigma<2^\rho N$을 만족하고, 임의의 허용 함수 f에 대해 $Auth$ 알고리즘에 의해 계산된 σ_1, …, σ_l을 대입하면 허용 함수의 정의에 의해 $0\leq f(\sigma_1,\cdots,\sigma_l)\leq 2^{bd(f)}\leq 2^\beta$가 성립하고, 따라서 $\sigma'=f(\sigma_1,\cdots,\sigma_l)\in\Sigma$이 성립한다. 즉, 적법하게 생성된 모든 MAC들이 MAC 공간 Σ 안에 들어가고, 또한 그러한 임의의 MAC σ는 $0\leq\sigma\leq 2^\beta$를 만족하므로, 그 비트 길이가 안전성 파라미터의 어떤 다항식 이하가 된다. □

정리 8. 이상의 준동형 MAC Π는 올바르다.

증명. 우선, MAC σ를 $\sigma\leftarrow Auth(sk,\tau,m)$로 계산하면 $Ver(sk,(id,\tau),m,\sigma)=1$이 성립함을 확인하자. 값 a를 $a=N^{-1}(r-m)\bmod p$, 그리고 MAC σ를 $\sigma=(pq+a)N+m$로 정의했으므로, $\sigma\equiv m\pmod N$이 성립하고, 또한 modulus p에 대해서는 $\sigma\equiv aN+m\equiv(r-m)+m\equiv r\pmod p$가 성립함을 알 수 있다.

이제 $\sigma_i\leftarrow Auth(sk,\tau_i,m_i)$로 각각의 MAC을 계산한 뒤, $\sigma'\leftarrow Eval(ek,f,\sigma_1,\ldots,\sigma_l)$로 계산했다고 하자. 그러면 정의에 의해 $\sigma'=f(\sigma_1,\cdots,\sigma_l)$이다. 허용 함수 f는 정수 계수 다항식이므로,

$$\begin{aligned}\sigma'\bmod N&\equiv f(\sigma_1\bmod N,\cdots,\sigma_l\bmod N)\\&\equiv f(m_1,\cdots,m_l)\pmod N\end{aligned}$$

그리고 마찬가지로, $r_i=F(k,\tau_i)$이면,

$$\begin{aligned}\sigma'\bmod p&\equiv f(\sigma_1\bmod p,\cdots,\sigma_l\bmod p)\\&\equiv f(r_1,\cdots,r_l)\pmod p\end{aligned}$$

따라서 이 경우

$$Ver(sk,(f,\tau_1,\ldots,\tau_l),f(m_1,\cdots,m_l),\sigma')=1$$

이 성립한다. □

IV. 안전성 분석

보조정리 9. 양의 정수 p, n에 대해, X를 $\mathbf{Z}_n$ 상에서 균일 분포를 갖는 확률변수라 하고, Y를 $\mathbf{Z}_p$ 상에서 균일 분포를 갖는 확률변수라 하면, $X\bmod p$와 Y 사이의 통계적 거리(statistical distance)는 최대 $p/(4n)$이다.

증명. n을 p로 나누어서 $n=pa+b$라고 놓자. 각각의 $k\in\mathbf{Z}_p$에 대해, $\Pr[X\bmod p=k]$를 계산하자. X의 가능한 값을 $pq+r$라 놓으면, 순서쌍 (q,r)은 총 n개이지만, $k\geq b$인 경우에는 $r=k$인 순서쌍은 $0\leq q<a$를 만족해야 하므로 총 a개, 그리고 $0\leq k<b$인 경우에는 $r=k$인 순서쌍은 $0\leq q\leq a$를 만족해야 하므로 총 $a+1$개다. 즉, $\Pr[X\bmod p=k]$는 $k\geq b$일 때 a/n, $0\leq k<b$일 때 $(a+1)/n$이 된다. 한편 임의의 $k\in\mathbf{Z}_p$에 대해 $\Pr[Y=k]=1/p$임은 분명하므로, 통계적 거리의 공식에 이들을 대입하고 정리하면 쉽게 $\triangle(X\bmod p,Y)\leq\frac{p}{4n}$을 얻을 수 있다. □

보조정리 10. 앞에서 묘사된 준동형 MAC Π는 메시지 유일성을 만족한다.

증명. 임의의 sk, $(f,\tau_1,...,\tau_l)$, m, m', σ에 대해, 만일 $Ver(sk,(f,\tau_1,...,\tau_l),m,\sigma)=1$와 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma)=1$이 동시에 성립한다고 가정하자. Ver 알고리즘의 정의에 의해, 이 경우 $\sigma \equiv m \pmod N$와 $\sigma \equiv m' \pmod N$이 둘 다 성립하므로, 결국 $\mathbf{Z}_N$의 원소인 m, m'은 같을 수밖에 없다. □

정리 11. 앞에서 묘사된 준동형 MAC Π는 MCT, 즉 MAC에 대한 상수성 검사 가능성을 만족한다.

증명. 준동형 MAC Π의 evaluation key ek를 하나 고정하자. Arity가 l인 허용 함수 f와, 집합 $\{1,\cdots,l\}$의 부분집합 I, 그리고 특정 메시지들의 순서쌍 $(m_i)_{i\in I}$, 그에 대응되는 MAC들의 순서쌍 $(\sigma_i)_{i\in I}$가 주어졌을 때에, $Eval(f,(\sigma_i)_{i\in I})$가 상수인지의 여부를 검사하는 효율적인, 그리고 무시할 만한 오류 확률을 갖는 확률적 알고리즘이 필요하다. 하지만 $e=Eval(f,(\sigma_i)_{i\in I})$는 정수 계수 다변수 다항식이고, 또한 허용 함수의 정의에 의해 $\deg(e)\le \deg(f)\le bd(f)\le \beta$로, 차수가 안전성 파라미터의 다항식이므로, 잘 알려진 Schwartz-Zippel 보조정리에 의한 polynomial identity testing이 가능하고 (예를 들어 [4] 참고), 이를 통해 e가 상수함수인지의 여부를 압도적인 확률로 판정할 수 있다. □

보조정리 12. 어떤 준동형 MAC Π가 메시지 유일성을 만족한다고 하자. 만일 A가 Π의 SUF 안전성을 공격하는 임의의 PPT 공격자라고 하면, A가 강한 위조를 성공시키는 경우에 이는 반드시 제 1형 강한 위조 혹은 제 2형 강한 위조이지, 단순한 위조일 수 없다.

증명. $((f,\tau_1,...,\tau_l),m',\sigma')$가 공격자 A의 성공적인 강한 위조라고 하고, 이것이 제 1종과 제 2종의 강한 위조가 아니라고 가정하자. 그렇다면 우선 $Ver(sk,(f,\tau_1,...,\tau_l),m',\sigma')=1$이 성립하고, 또한 각각의 $i=1,\cdots,l$ 중에서 $(\tau_i,m_i,\sigma_i)\in S$를 만족하는 m_i,σ_i가 존재하는 i들의 집합을 I라고 하면 $Eval(f,(\sigma_i)_{i\in I})$은 상수함수이며 (1종이 아니므로), 또한 그 함수값은 σ'이 된다 (2종이 아니므로). Π는 또한 메시지 유일성을 만족하므로, 보조정리 5에 의해 $App(f,(m_i)_{i\in I})$ 또한 상수함수이다. 그 상수값을 m^*라 하면, 각각의 $j\notin I$에 대해 임의로 m_j를 고르고 $\sigma_j \leftarrow Auth(sk,\tau_j,m_j)$를 계산하면, 가정에 의해 $\sigma'=Eval(ek,f,\sigma_1,\cdots,\sigma_l)$와 또한 $m^*=f(m_1,\cdots,m_l)$이 성립하고, Π가 올바르기 때문에 $Ver(sk,(f,\tau_1,...,\tau_l),m^*,\sigma')=1$이 성립하고, 그러면 메시지 유일성에 의해 $m^*=m'$이 성립하는데, 이 경우 $((f,\tau_1,...,\tau_l),m',\sigma')$는 성공적인 위조가 아니게 되며, 가정에서 제 1,2종 강한 위조도 아니므로 성공적인 강한 위조라는 가정과 모순된다. 따라서, $((f,\tau_1,...,\tau_l),m',\sigma')$가 공격자 A의 성공적인 강한 위조라면 이는 반드시 제 1종 혹은 제 2종이어야 한다. □

정리 13. 앞에서 묘사된 준동형 MAC Π는 SUF 안전성을 만족한다.

증명. A가 임의의 PPT 공격자라고 하자. 우선, 보조정리 12에 의하면 A의 강한 위조 중에서는 위조의 경우가 아니라 제 1종 강한 위조 및 제 2종 강한 위조만 고려하는 것으로 충분하다.

우리는 A가 참여하는 게임을 단계적으로 변형해나가는 방식으로 준동형 MAC Π의 안전성을 증명할 것이다. 우선 게임 G0을 Π에 대한 정상적인 SUF 게임이라고 놓자. 다음, 게임 G1은 G0에 사용된 PRF F를 실제 랜덤 함수 $\alpha:\{0,1\}^\lambda\to\{0,1\}^\rho$로 교체한 게임이라 하자. 즉, 예를 들어 $Auth(sk,\tau,m)$ 알고리즘에서 $r\leftarrow F(k,\tau)$를 계산하는 대신, G1에서는 $r\leftarrow\alpha(\tau)$로 계산한다. F가 안전한 PRF라고 가정하면, 공격자 A의 게임 G0에 대한 advantage와 G1에 대한 advantage의 차이는 무시할 만함을 알 수 있다. 게임 G1에서 공격자 A는 일련의 MAC 생성 질의 후에 최종적으로 위조 시도 $((f,\tau_1,...,\tau_l),m',\sigma')$를 출력하며, 이때 게임 G1의 출력값이 1이 되는 것은 정확히 이 위조 시도가 제 1종 강한 위조이거나 제 2종 강한 위조이거나의 경우이다. 따라서 $((f,\tau_1,...,\tau_l),m',\sigma')$에 대한 게임 G1의 출력 알고리즘은 다음과 같이 정리할 수 있다:

- 만일 $Eval(f,(\sigma_i)_{i\in I})$이 함수값 σ'을 갖는 상수함수이면 0을 출력
- 그렇지 않은 경우,

- 각각의 $i \in I$에 대해 $r_i \leftarrow \alpha(\tau_i)$로 정의하고
- 각각의 $j \notin I$에 대해 $r_j \leftarrow \alpha(\tau_j)$로 정의하고
- $r = f(r_1, \cdots, r_l)$을 계산한 뒤
- 만일 $\sigma' \equiv m' \pmod N$이고 또한 $\sigma' \equiv r \pmod p$이면 1을 출력, 그렇지 않은 경우 0을 출력

한편, 위에서 $i \in I$인 경우 $\sigma_i \equiv \alpha(\tau_i) \pmod p$이 성립하고, 또한 $j \notin I$의 경우 $\alpha(\tau_j)$는 $\{0,1\}^\rho$ 상의 균일한 난수값이므로, 위의 출력 알고리즘은 다음과 동치이다:

- 만일 $Eval(f, (\sigma_i)_{i \in I})$이 함수값 σ'을 갖는 상수함수이면 0을 출력
- 그렇지 않은 경우,
  - 각각의 $i \in I$에 대해 $r_i \leftarrow \sigma_i$로 정의하고
  - 각각의 $j \notin I$에 대해 $r_j \leftarrow \{0,1\}^\rho$로 정의
  - $r = f(r_1, \cdots, r_l)$을 계산한 뒤
  - 만일 $\sigma' \equiv m' \pmod N$이고 또한 $\sigma' \equiv r \pmod p$이면 1을 출력, 그렇지 않은 경우 0을 출력

이렇게 표현하면, G1에서 랜덤 함수 $\alpha : \{0,1\}^\lambda \rightarrow \{0,1\}^\rho$는 단지 MAC 생성 질의에만 등장한다. 이제 게임 G2에서는 MAC 생성 질의를 다음과 같이 변형한다:

MAC 생성 질의: 공격자 A가 MAC 생성 질의 (τ, m)을 하면, $q' \leftarrow [0, 2^\rho) \cap \mathbf{Z}$를 뽑고, $\sigma = q'N + m$을 계산한 뒤, $S \leftarrow S \cup \{(\tau, m, \sigma)\}$로 갱신하고, σ를 질의에 대한 답으로 반환한다.

게임 G2에서는 랜덤 함수 α가 등장하지 않을 뿐만 아니라, MAC 생성 질의의 과정에서 소수 p 역시 등장하지 않는다. 그렇다면 게임 G2에서 소수 p는 공격자 A가 위조 시도를 하고 난 뒤에 최종적으로 게임의 출력이 결정되는 마지막 단계에만 등장한다. 따라서, 이 η 비트 소수 p를 이 마지막 단계에 선택해도 무방하다. 이를 게임 G3으로 정의하자. G3에서는 공격자의 위조 시도 후의 출력 알고리즘은 다음과 같다:

- 만일 $Eval(f, (\sigma_i)_{i \in I})$이 함수값 σ'을 갖는 상수함수이면 0을 출력
- 그렇지 않은 경우,
  - 각각의 $i \in I$에 대해 $r_i \leftarrow \sigma_i$로 정의하고
  - 각각의 $j \notin I$에 대해 $r_j \leftarrow \{0,1\}^\rho$로 정의
  - $r = f(r_1, \cdots, r_l)$을 계산한 뒤
  - η 비트의 랜덤한 소수 p를 선택하고
  - 만일 $\sigma' \equiv m' \pmod N$이고 또한 $\sigma' \equiv r \pmod p$이면 1을 출력, 그렇지 않은 경우 0을 출력

이제 각각의 게임을 비교해보자. 게임 G0과 G1에서 공격자의 이득의 차이는 F가 안전한 PRF이기 때문에 무시할 만하다. 게임 G1과 G2의 차이는 MAC 생성 질의의 답변에 있다. G1에서는 $a = N^{-1}(\alpha(\tau) - m) \bmod p$를 계산한 뒤 $q \leftarrow [0, \lfloor 2^\rho / p \rfloor) \cap \mathbf{Z}$를 선택하고, $\sigma = (pq + a)N + m$를 계산하는데, 이때 보조정리 8에 의하면 $\alpha(\tau) \bmod p$의 분포와 $\mathbf{Z}_p$ 상의 균일 분포와의 통계적 거리는 다음을 만족한다.

$$\triangle \le \frac{p}{4 \cdot 2^\rho} < \frac{2^\eta}{4 \cdot 2^\rho} = \frac{1}{2^{\rho - \eta + 2}}$$

따라서 a의 분포 역시 $\mathbf{Z}_p$ 상의 균일 분포와 최대 $2^{-(\rho-\eta+2)}$만큼의 통계적 거리를 갖는다. $\rho = \eta + \omega(\lg \lambda)$이므로 $2^{-(\rho-\eta+2)}$는 무시할 만하고, 그러므로 a가 $\mathbf{Z}_p$ 상에서 균일하게 뽑은 난수라고 놓아도 좋다. 이 경우 $pq + a$는 $\{0,1,\cdots,p \lfloor 2^\rho / p \rfloor - 1\}$ 상의 균일 분포를 따르고, 게임 G2에서의 q'는 $\{0,1,\cdots,2^\rho - 1\}$ 상의 균일 분포를 따른다. 간단한 계산에 의해 이제 $pq + a$와 q'의 통계적 거리는 최대 $\frac{p}{2^\rho} < \frac{2^\eta}{2^\rho} = \frac{1}{2^{\rho-\eta}}$이고, 다시 $\rho = \eta + \omega(\lg \lambda)$에 의해 이는 무시할 만함을 알 수 있다. 그러므로 게임 G1과 G2의 전체적인 차이 역시 무시할 만하다.

마지막으로 게임 G3은 게임 G2에서 소수 p의 선택만 실제로 필요한 순간까지 미룬 것에 불과하므로, 둘 사이의 차이는 없다.

이제 게임 G3의 출력이 1이 될 확률이 무시할 만함을 증명하기만 하면 된다. 먼저, $e := Eval(f, (\sigma_i)_{i \in I})$이 함수값 σ'을 갖는 상수함수가 아닌 경우, $r = f(r_1, \cdots, r_l) = \sigma'$일 확률이 무시할 만함을 보이도록 하자.

우선, $Eval(f, (\sigma_i)_{i \in I})$이 함수값 σ^*을 갖는 상수함수이지만 $\sigma^* \neq \sigma'$인 경우에는, 정의에 의해 $f(r_1, \cdots, r_l) = e(r_j)_{j \notin I} = \sigma^* \neq \sigma'$이므로, 이 경우에

$r = f(r_1, \cdots, r_l) = \sigma'$일 확률은 0이다.

다음, $e = Eval(f, (\sigma_i)_{i \in I})$이 상수함수가 아닌 경우를 생각하자. 이 경우, 무작위로 $r_j \leftarrow \{0,1\}^\rho$들을 선택할 때에 $f(r_1, \cdots, r_l) = e(r_j)_{j \notin I} = \sigma'$일 확률은 Schwartz-Zippel 보조정리에 의해 $\deg(f)/2^\rho$ 이하임을 알 수 있고, 이 경우에도 무시할 만하다.

게임 G3의 출력이 1인 경우에는 $Eval(f, (\sigma_i)_{i \in I})$이 함수값 σ'을 갖는 상수함수가 아니고 또한 $\sigma' \equiv r \pmod p$이 성립한다. 이 경우 무시할 만한 확률을 제외하고는 $\sigma' \neq r$라고 가정할 수 있다. 하지만 아래의 보조정리 14를 이용하면, $\sigma' \neq r$인 경우에 균일하게 무작위로 선택한 η 비트 소수 p에 대해 $\sigma' \equiv r \pmod p$이 성립할 확률은

$$\frac{(\ln 2)\lg|\sigma' - r|}{2^{\eta-2}} \leq \frac{(\ln 2)\lg(|\sigma'| + |r|)}{2^{\eta-2}}$$
$$\leq \frac{(\ln 2)\lg(2^{\beta+1})}{2^{\eta-2}} \leq \frac{(\ln 2)(\beta+1)}{2^{\eta-2}}$$

로, 역시 무시할 만하다. (앞에서 $\sigma' \in \Sigma$이므로 $|\sigma'| \leq 2^\beta$, 그리고 $f \in F$이고 각 $0 \leq r_k \leq 2^\rho N$이므로 $|r| = |f(r_1, \cdots, r_l)| \leq 2^{bd(f)} \leq 2^\beta$ 성립.)

따라서, 게임 G3의 출력이 1일 확률은 무시할 만하고, 여태까지를 종합하면 공격자 A가 성공적인 강한 위조를 출력할 확률 역시 무시할 만하므로, 준동형 MAC Π는 SUF 안전성을 만족한다. □

보조정리 14. 임의의 양의 정수 m이 주어졌을 때, 균일하게 무작위로 선택한 η 비트 소수 p가 m을 나눌 확률은 최대 $\frac{(\ln 2)\lg(m)}{2^{\eta-2}}$이다.

증명. m을 소인수분해했을 때 나타나는 서로 다른 η 비트 소수가 모두 t개 있다고 하고, 이들을 $p_1, \cdots, p_t$로 나타내자. 그러면

$(2^{\eta-1})^t \leq p_1 p_2 \cdots p_t \leq m$이므로, $t \leq \frac{\lg(m)}{\eta-1}$이 성립한다. 그렇다면, 균일하게 무작위로 선택한 η 비트 소수 p가 m을 나눌 확률은 정확히 $t/|P|$인데, 여기서 P는 모든 η 비트 소수들의 집합을 의미한다. $|P|$는 소수 정리를 이용하여 추산할 수 있는데, $\eta \geq 4$이기만 하면 $|P| \geq \frac{2^{\eta-2}}{(\eta-1)\ln 2}$임을 얻을 수 있다 ([5], p. 524). t와 $|P|$에 대한 부등식을 결합하면, 확률은 최대 $\frac{\lg(m)}{\eta-1} \cdot \frac{(\eta-1)\ln 2}{2^{\eta-2}} = \frac{(\ln 2)\lg(m)}{2^{\eta-2}}$임을 얻는다. □

따름정리 15. 준동형 MAC Π는 SUFV 안전성을 만족하고, 따라서 UFV 안전성을 만족한다.

증명. 이는 보조정리 10, 정리 11, 정리 13, 정리 6, 그리고 정리 2에 의한 결과이다. 메시지 유일성과 MAC에 대한 상수성 검사 가능성(MCT), 그리고 SUF 안전성을 만족하므로, 따라서 SUFV 안전성과 UFV 안전성을 만족한다. □

V. 효율성에 관한 논의

여기에서는 본 논문에서 제시한 준동형 MAC의 효율성에 대한 분석을 하고자 한다. 우선 준동형 MAC의 검증 알고리즘에 관한 일반적인 효율성 관련 논의를 하고, 그 다음 논문[2]의 Catalano -Fiore 방식과 본 논문의 방식의 효율성을 간략히 비교할 것이다.

[1]에서 논의된 바와 같이, 본 논문의 준동형 MAC을 포함하여 기존의 준동형 MAC의 거의 전부[1,2]의 경우, 검증 계산의 효율성이 해당 함수를 직접 계산하는 것에 비해 더 낫지 않다. 이는 준동형 서명을 포함하여도 마찬가지이다[7]. 따라서, 클라우드 컴퓨팅으로의 응용에서 사용자의 관점에서 볼 때, 이러한 방식은 특정 함수를 계산하는 계산 복잡도 면에서 이득을 주기보다는, 함수 계산을 위해 필요한 입력을 전송하는 통신 복잡도의 관점에서 이득을 준다: 즉, 예를들어 사용자가 저장을 위탁한 데이터가 총 1GB인데, 그중 특정 함수 f의 계산을 위해 입력 10MB가 필요하고, f의 함수값의 크기가 10바이트라고 한다면, 준동형 MAC 혹은 서명이 사용되지 않는 상황에서라면 10MB의 데이터 및 그와 연관된 전통적인 MAC을 내려받아서, MAC 검증을 통해 해당 데이터가 원본과 다르지 않음을 확인한 뒤, 사용자가 직접 f를 계산해야 하는데 비해, 준동형 MAC을 사용하는 경우에는 f의 최종 결과값인 10바이트 및 그에 대한 MAC 값만을 받은 뒤, 검증 계산을 통해 결과값이 올바름을 확인하면 되므로 통신 복잡도 면에서 현저한 개선을 얻을 수 있으나, 검증 계산의 효율성이 함수의 직접 계산에 비해 더 낫지 않기 때문에 사용자의 계산 능력이 약한 환경에 적용하기는 어렵다.

여태까지 알려진 유일한 예외적인 경우는 [6]의 준동형 MAC이다. 이들의 준동형 MAC은 검증 질의를 허용해도 안전성을 유지하며, 무엇보다도 다른 준동형 MAC/서명들과는 달리 'amortized model'하에서 검증 계산이 함수의 직접 계산에 비해 더 나은 계산 복잡도를 갖는다. 다만, [1,2]의 준동형 MAC, 그리고 본 논문의 준동형 MAC과는 달리, 단지 최대 2차식까지의 함수들만을 허용 함수로 갖는다는 제약을 갖는다. 이러한 차수에 대한 제약이 없는 일반적인 경우에서 효율적인 검증 알고리즘을 갖는 준동형 MAC을 설계하는 것은 중요한 미해결 문제로, 향후 이에 관한 연구를 계속 진행하도록 할 것이다.

Catalano-Fiore의 준동형 MAC[2]은 다른 알려진 준동형 서명 방식이나 준동형 암호화 방식들과는 달리 복잡한 안전성 가정에 기반하지 않고 있다. 다만, 사용된 PRF가 안전해야 하고, modulus p가 충분히 커서 $1/p$가 무시할만큼 작아야(negligible) 한다는 두 가지 조건만 만족하면 안전성이 증명된다. 하나의 MAC은 $\mathbf{Z}_p$에서 계수를 갖는 다항식이고, 특히 $Auth$ 알고리즘으로부터 직접 생성된 MAC은 1차식이다. $Eval$ 알고리즘은 다항식의 단순한 덧셈 혹은 곱셈 연산으로 구성되어 있어서, 곱셈의 횟수에 비례하여 MAC의 크기가 증가하게 된다. 구체적으로, p의 비트 길이를 안전성 파라미터 λ로 잡으면, $Auth$ 알고리즘이 출력하는 1차식 MAC의 길이는 $O(\lambda)$가 되고, 곱셈을 여러번 하여 차수가 d로 증가하면, MAC의 길이는 $O(d\lambda)$이 된다.

본 논문에서 제시된 준동형 MAC 역시, 복잡한 안전성 가정에 의존하지 않고, Catalano-Fiore의 경우와 마찬가지로 사용된 PRF가 안전하고, 소수 p와 난수 q의 크기가 적절하게 되어 있는 것으로 충분하다. 정리 7에 의해, $Auth$ 알고리즘이 출력하는 MAC의 길이는 최대 $\rho+\eta=O(\lambda)$ 비트이고, 본 논문의 준동형 MAC의 경우에는 MAC은 다항식이 아니라 정수로 주어지므로, 곱셈 연산을 d회 하면 MAC의 비트 길이는 $O(d\lambda)$로 증가하게 된다. 근본적으로, Catalano-Fiore MAC의 경우 유한체 상의 다항식 계산에, 본 논문의 MAC은 정수의 계산에 기반하고 있다는 차이가 있으나, 둘 다 비슷한 크기의 MAC을 생성하고 처리한다.

VI. 결론

본 논문에서는 새로운 준동형 MAC을 제시하고, 그 안전성을 증명하였다. 본 논문의 준동형 MAC은 Catalano와 Fiore의 방식과 비교할 만한 충분한 실용적인 효율성을 제공하며, 또한 강한 위조 불가능성 개념을 이용하여, 검증 질의를 허용하는 모델에서의 안전성을 만족시킴을 보였다. 이 분야에서 중요한 미해결 문제는 완전 준동형성을 만족하는 준동형 MAC을 설계하는 것인데, 그 안전성 증명에서 향후 강한 위조 불가능성은 유용한 도구가 될 수 있을 것으로 생각한다.

References

[1] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," Advances in Cryptology, ASIACRYPT 2013, LNCS 8270, pp. 301-320, 2013.

[2] D. Catalano and D. Fiore, "Practical homomorphic MACs for arithmetic circuits," Advances in Cryptology, EUROCRYPT 2013, LNCS 7881, pp. 336-352, 2013.

[3] M. Bellare, O. Goldreich and A. Mityagin, "The power of verification queries in message authentication and authenticated encryption," IACR ePrint 2004-309, Nov. 2004.

[4] S. Arora and B. Barak, Computational complexity: a modern approach, Cambridge University Press, 2009.

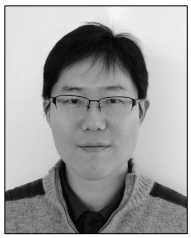
[5] J. von zur Gathen and J. Gerhard, Modern computer algebra, 3rd Ed., Cambridge University Press, 2013.

[6] M. Backes, D. Fiore and R.M. Reischuk, "Verifiable delegation of computation on outsourced data," Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13), pp. 863-874, Nov. 2013.

[7] D. Boneh and D.M. Freeman, "Homomorphic signatures for polynomial functions," Advances in Cryptology, EUROCRYPT 2011, LNCS 6632, pp. 149-168, 2011.

〈저자소개〉

주 치 홍 (Chihong Joo) 학생회원
2007년 2월: 서울대학교 수학과 졸업
2011년 2월: 서울대학교 수학과 석사
2011년 3월~현재: 울산과학기술대학교 박사과정
〈관심분야〉 암호 이론, 컴퓨터 보안



윤 아 람 (Aaram Yun) 정회원
1995년 2월: 한국과학기술원 학사과정 졸업
2001년 6월: Yale University 수학 박사
2001년 ~ 2003년: 삼성 SDS 정보기술연구소 과장
2003년 ~ 2007년: ETRI 부설 국가보안기술연구소 선임연구원
2007년 ~ 2010년: University of Minnesota 박사후 연구원
2010년 ~ 현재: 울산과학기술대학교 조교수
〈관심분야〉 암호 이론, 컴퓨터 보안