국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

보안과제( ), 일반과제(O), NIMS-2016-01-0002
주요사업 연차보고서

사물인터넷 보안을 위한 암호 핵심기술 개발

(Development of Public-Key Cryptographic Algorithms for Internet of Things)

2017. 1

국가수리과학연구소

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 출 문

국가수리과학연구소장 귀하

본 보고서를 “사물인터넷 보안을 위한 암호 핵심기술 개발”사업의 연차보고서로 제출합니다.

2017. 01. 31.

주관연구기관명 : 국가수리과학연구소

주관연구책임자 : 심 경 아

참 여 연 구 원 : 박 철 민

참 여 연 구 원 : 구 남 훈

참 여 연 구 원 : 김 애 영

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

보고서 요약서

<table>
<tr><td>과제고유번호</td><td>A21200000</td><td>해당단계 연구기간</td><td>2016.01.01.-
2016.12.31</td><td>단계구분</td><td>1(해당단계)/
3(총단계)</td></tr>
<tr><td>연구사업명</td><td colspan="5">사물인터넷 보안을 위한 암호 핵심기술 개발</td></tr>
<tr><td>연구과제명</td><td colspan="5">-</td></tr>
<tr><td rowspan="2">연구책임자</td><td rowspan="2">심경아</td><td>해당단계 참여 연구원 수</td><td>총: 4 명
내부: 4 명
외부: 명</td><td>해당단계 연구비</td><td>정부: 420,000 천원
민간: 천원
계: 천원</td></tr>
<tr><td>총 연구기간 참여 연구원 수</td><td>총: 명
내부: 명
외부: 명</td><td>총 연구비</td><td>정부: 천원
민간: 천원
계: 천원</td></tr>
<tr><td>연구기관명 및 소속부서명</td><td colspan="2">국가수리과학연구소
융합수학연구부</td><td>참여기업명</td><td colspan="2">-</td></tr>
<tr><td>협동연구</td><td colspan="3">연구기간명 :</td><td colspan="2">연구책임자 :</td></tr>
<tr><td>위탁연구</td><td colspan="3">연구기관명 :</td><td colspan="2">연구책임자 :</td></tr>
</table>

○ 고속 공개키 암호알고리즘 설계에 가장 적합한 난제 선택을 위한 다양한 난제 기반 공개키 암호 설계방법, 요구조건, 문제점, 효율성 비교 분석 및 도전과제 도출
- 격자기반, 다변수 이차식 기반, 코드기반, 해쉬 기반 공개키 암호알고리즘 설계 방법, 요구조건, 문제점 비교 분석
- 격자기반, 다변수 이차식 기반, 코드기반, 해쉬 기반 공개키 암호알고리즘 8-비트, 16-비트, 32-비트 플랫폼에서의 효율성 비교 분석
- 리소스가 제한된 경량기기에서의 실제 구현을 위해 해결해야할 도전과제 도출

○ 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석 및 설계원칙 도출
- 기존에 설계된 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석
- 다양한 공격에 대한 안전성 분석을 통한 설계원칙 도출
- MQ-문제 기반 identification protocol을 Fiat-Shamir의 변환을 통해 얻은 전자서명의 안전성 분석

○ 다변수 이차식 기반 고속 공개키 암호알고리즘 설계 및 안전성 분석
☞ 짧은 키 길이를 갖는 다변수 이차식 기반 고속 공개키 암호알고리즘 설계
☞ 알려진 공격에 대한 안전성 분석

색인어 (각 5개 이상)	한글	수학적 난제, 공개키 암호알고리즘, 전자서명 알고리즘, 다변수 이차식
	영어	Mathematically hard problem, Public-Key Cryptography, Digital Signature, Multivariate Quadratic Equations

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

〈 국문 요약문 〉

<table>
<tr><td>연구의
목적 및 내용</td><td colspan="5">○ 연구 목표
- 신규 수학적 난제에 기반한 고속 공개키 암호알고리즘 개발
○ 연구 내용
- 다양한 난제 기반 공개키 암호 설계방법, 요구조건, 문제점, 효율성 비교 분석 및 도전과제 도출
- 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석 및 설계원칙 도출
- 다변수 이차식 기반 고속 공개키 암호알고리즘 설계 및 안전성 분석
- 설계한 고속 공개키 암호알고리즘 구현 및 효율성 비교 분석</td></tr>
<tr><td>연구개발성과</td><td colspan="5">○ 다양한 난제 기반 공개키 암호 설계방법, 요구조건, 문제점, 효율성 비교 분석 결과
○ 기존 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석 결과
○ 다변수 이차식 기반 고속 공개키 암호알고리즘 설계 및 안전성 분석 결과</td></tr>
<tr><td>연구개발성과의
활용계획
(기대효과)</td><td colspan="5">○ 개발된 고속 공개키 암호는 IoT 경량기기에서 활용가능하며, 다양한 응용환경에서 요구하는 특수한 목적과 기능을 수행할 수 있는 암호 응용기술에 기반이 되는 원천기술로 활용될 것임.
○ 다변수 이차식으로 구성된 시스템의 해를 구하는 문제는 양자 컴퓨터에 안전한 문제로 알려져 있으며, 연산이 현재 사용하고 있는 공개키 암호 RSA나 ECC처럼 큰 수의 modular reduction이나 타원곡선 군의 복잡한 연산 대신 작은 유한체의 곱셈만이 요구되므로 속도 면에서 기존 암호알고리즘보다 월등히 빨라 경쟁력 있는 기술 확보가 가능함.
○ 개발된 고속 공개키 암호는 국제 표준화를 통해 세계적으로 국산 암호알고리즘을 사용할 수 있도록 함으로써, 세계 정보보호시장을 선점하고 국부 창출에 기여할 것임.
○ 개발된 고속 공개키 암호를 이용한 암호 응용기술들은 다양한 분야에서의 프라이버시 침해 문제 해결에 기여할 것이며, 정보보호에 필요한 인력양성과 IoT 관련 보안산업의 활성화에 따른 경제적 효과를 기대할 수 있음.</td></tr>
<tr><td>핵심어
(5개 이내)</td><td>공개키
암호알고리즘</td><td>전자서명
알고리즘</td><td>다변수 이차식
시스템</td><td>수학적 난제</td><td>사물인터넷</td></tr>
</table>

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

〈 SUMMARY 〉

Purpose & Contents	○ Objective - Development of practical public-key cryptographic algorithms based on new mathematical hard problems ○ Contents - Analysis of security and efficiency of public-key cryptographic algorithms based on various mathematical hard problems - Analysis of public-key cryptographic algorithms based on multivariate quadratic equations and deduction of their design principles - Design of a public-key signature scheme based on multivariate quadratic equations and its security analysis
Results	○ Analysis results of public-key cryptographic algorithms based on various mathematical hard problems ○ Security analysis results of existing public-key cryptographic algorithms based on multivariate quadratic equations ○ A new practical public-key signature scheme based on multivariate quadratic equations and its security analysis result
Expected Contribution	○ The practical public-key cryptographic algorithm requires only modest computational resources that makes them attractive for the use of low-cost devices such as smart cards. We believe that our scheme is a leading candidate for security of these constrained devices. ○ This research will contribute to secure fundamental technology for Internet of Things in Korea, and take the lead in the international standards. ○ The proposed efficient special-purposed cryptographic algorithms will contribute to dominate the global market in advance, and thereby increase national wealth through international standardization. ○ The proposed practical cryptographic algorithm can be applied to support secure communication and authentication among the resource-constrained IoT devices.

Keywords	Public-key cryptographic algorithm	Public-key signature scheme	Multiariate quadratic equations	Mathematical hard problem	Internet of Things (IoT)

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

〈 CONTENTS 〉

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

〈 목 차 〉

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

표 목차

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

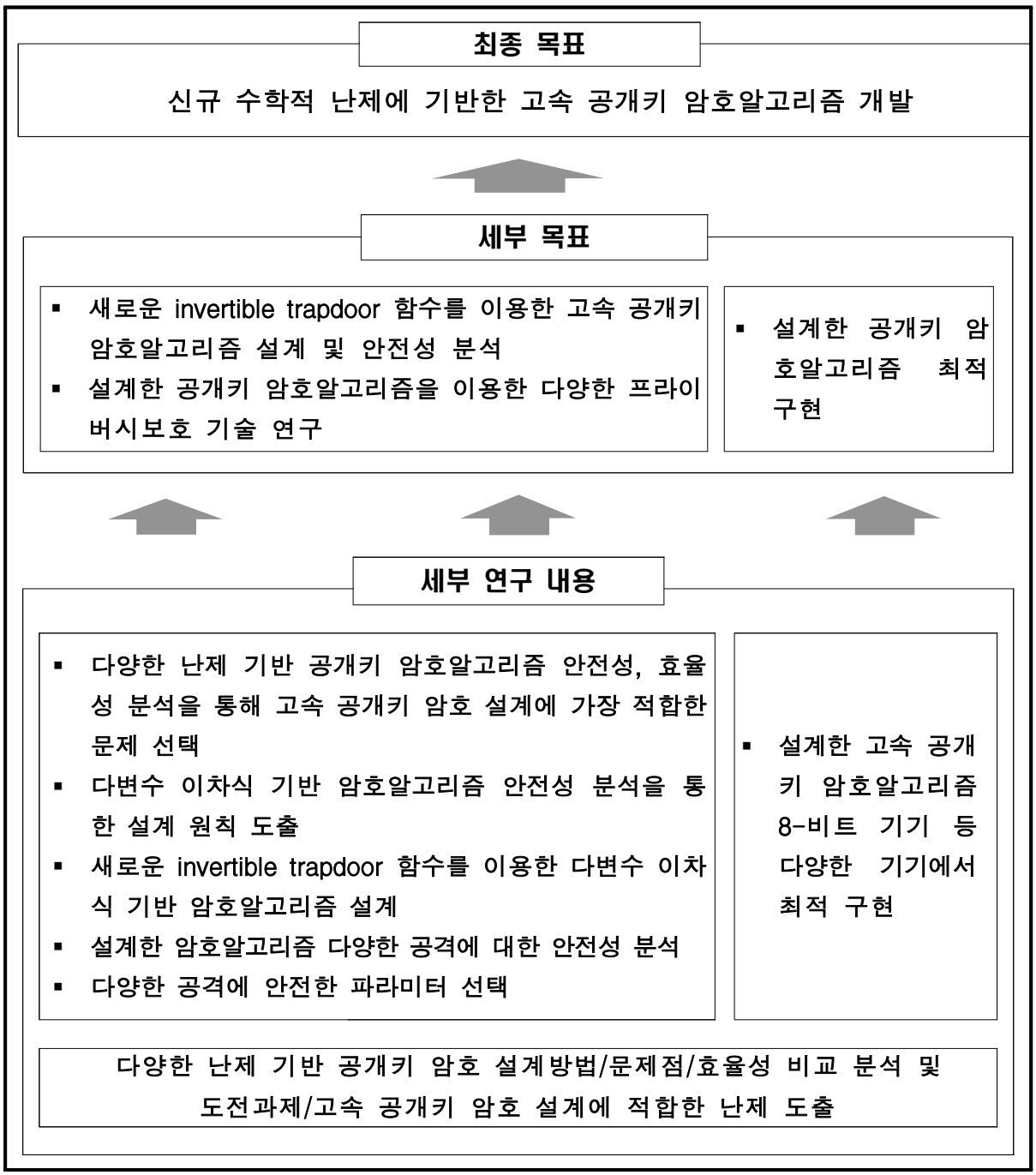
그림 목차

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 1 장. 연구개발과제의 개요

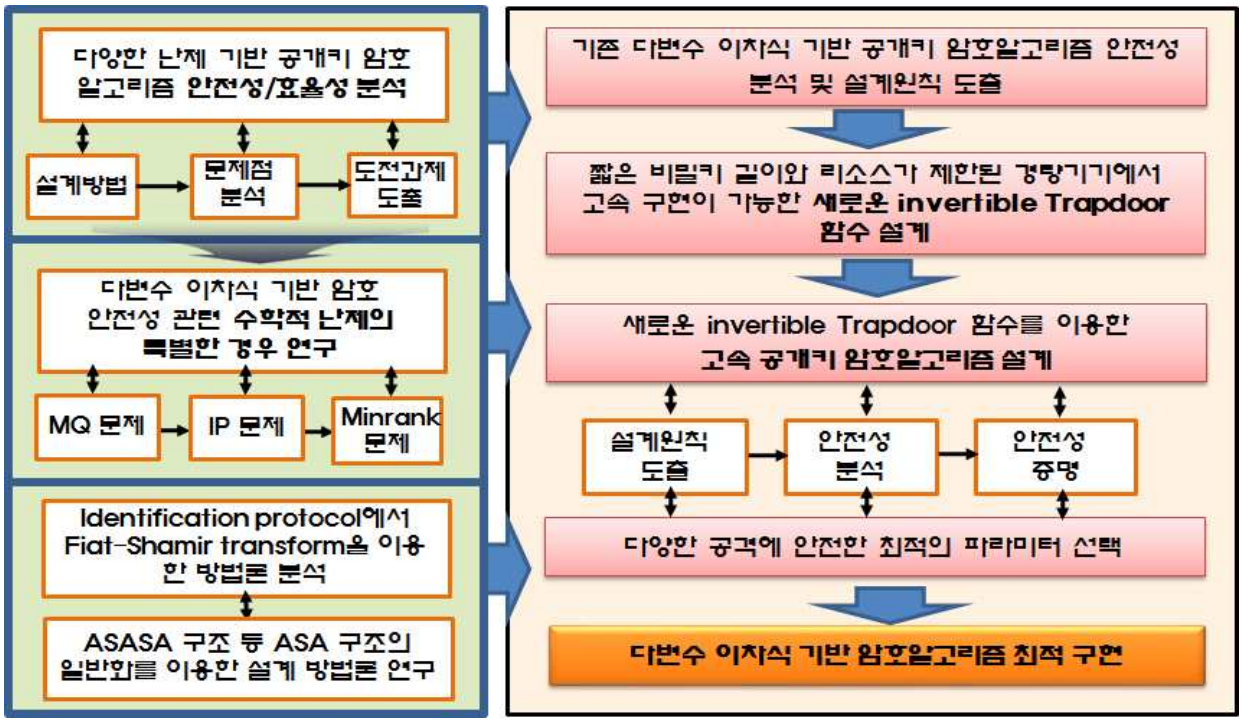
제 1 절. 연구개발 목표

신규 수학적 난제를 이용하여 리소스가 제한된 경량기기에서 고속으로 구현 가능한 고속 공개키 암호알고리즘을 설계하고 안전성 분석을 목적으로 한다.



국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 2절. 연구개발 내용 및 범위



○ 연구 내용

- 고속 공개키 암호알고리즘 설계에 가장 적합한 난제 선택을 위한 다양한 난제 기반 공개키 암호 설계방법, 요구조건, 문제점, 효율성 비교 분석 및 도전과제 도출
  - ☞ 격자기반, 다변수 이차식 기반, 코드기반, 해쉬 기반 공개키 암호알고리즘 설계방법, 요구조건, 문제점 비교 분석
  - ☞ 격자기반, 다변수 이차식 기반, 코드기반, 해쉬 기반 공개키 암호알고리즘 8-비트, 16-비트, 32-비트 플랫폼에서의 효율성 비교 분석
  - ☞ 리소스가 제한된 경량기기에서의 실제 구현을 위해 해결해야할 도전과제 도출
- 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석 및 설계원칙 도출
  - ☞ MQ-문제와 IP-문제를 이용한 다변수 이차식 기반 공개키 암호알고리즘 안전성 분석
    · direct 공격, 키 복구 공격, rank 분석을 통한 다양한 공격에 대한 안전성 분석
  - ☞ 다양한 공격에 대한 안전성 분석을 통한 설계원칙 도출
  - ☞ IP-문제의 이용 없이 MQ-문제로의 안전성 귀착을 갖는 다변수 이차식 기반 전자서명 알고리즘 안전성 분석
    · MQ-문제 기반 identification protocol을 Fiat-Shamir의 변환을 통해 얻은 전자서명의 안전성 분석
- 다변수 이차식 기반 고속 공개키 암호알고리즘 설계 및 안전성 분석
  - ☞ 짧은 키 길이를 갖는 다변수 이차식 기반 고속 공개키 암호알고리즘 설계
  - ☞ 알려진 공격에 대한 안전성 분석

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 2 장. 국내외 기술 개발 현황

제 1 절. 다양한 난제 기반 공개키 암호알고리즘 효율성 비교 분석

이 절에서는 다양한 수학적 문제에 기반을 둔 공개키 암호알고리즘의 8-비트, 16-비트, 32-비트 기기에서 구현한 결과들로부터 효율성을 비교 분석한다.

1. 타원곡선 이산대수문제 기반 공개키 암호알고리즘

현재 무선 통신에서 주로 사용되고 있는 공개키 암호알고리즘 타원곡선 암호(ECC)는 타원곡선 이산대수문제에 기반을 두고 있다. 다음 표들은 NIST에서 표준으로 제정된 타원곡선 NIST P-256과 Curve 22519에서의 상수 배 (scalar multiplication) 연산을 8-비트, 16-비트, 31-비트 기기에서 구현한 결과를 clock cycle로 나타낸 것이다. 표에 이용된 타원곡선의 상세는 다음과 같다.

- NIST P-256: 128-비트 안전도로 modulo prime을 $2^{256}-2^{224}+2^{192}+2^{96}-1$로 갖는 NIST 표준 타원곡선
- Curve 22519: Twisted Edwards curve인 Ed25519와 birationally equivalent한 타원곡선으로 2015년에 차세대 타원곡선 중의 하나로 채택되어 IETF internet standard draft 상태에 있음

표 1은 두 타원곡선에서의 상수 배 연산을 8-비트 AVR, ATmega2560, 256KB ROM, 8KB RAM에서 구현한 결과이다. 표 2는 16-비트 MSP 430X (8MHz) 64KB FRAM, 2 kB SRAM에서 구현한 결과이고, 표 3은 32-비트 ARM 64KB ROM, 8KB RAM에서 구현한 결과이다.

암호화 알고리즘	상수배 연산	RAM
NIST P-256 [WUW13]	34,930,000	590 Bytes
Curve25519 [DHHH15]	13,900,397	494 Bytes

표 1. 8-비트 AVR에서 타원곡선의 상수 배 연산을 구현한 결과

암호화 알고리즘	상수배 연산(Cycle)		Stack usage
NIST P-256 [GOL12]	16-bit H/W multiplier	7,284,377	n/a
	32-bit H/W multiplier	5,321,776	n/a
Curve25519 [DHHH15]	16-bit H/W multiplier	7,933,296	384 Bytes
	32-bit H/W multiplier	5,301,792	382 Bytes

표 2. 16-비트 MSP 430X (8MHz) 에서 타원곡선의 상수 배 연산을 구현한 결과

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

암호화 알고리즘	상수배 연산(Cycle)	RAM
NIST P-256 [WUW13]	10,730,000	540 Bytes
Curve25519 [DHHH15]	3,589,850	548 Bytes

표 3. 32-비트 ARM에서 타원곡선의 상수 배 연산을 구현한 결과

2. 격자 기반 공개키 암호알고리즘

격자 기반 공개키 암호알고리즘은 LWE 문제, Ring-LWE 문제, SVP 문제 등 격자를 이용한 다양한 문제에 기반을 두고 있다. 격자 기반 공개키 암호알고리즘은 homomorphic 성질 때문에 동형암호 (homomorphic encryption) 등 특별한 성질을 갖는 다양한 특수 암호알고리즘 설계에 널리 이용되고 있다. 현재 격자 기반 공개키 암호화 알고리즘 중에 Ring-LWE 기반 암호화 알고리즘이 가장 빠른 것으로 알려져 있다. 표 4의 결과는 32MHz, 128 kB flashable program memory, 8 kB SRAM을 갖는 8-비트 ATxmega128A1 processor와 NEON engine을 갖는 ARM Cortex-A9에서 128-비트 안전도의 Ring-LWE 기반 암호화 알고리즘을 구현한 결과이다.

암호화 알고리즘	Platform	키생성	암호화	복호화
Ring-LWE 기반 암호화 알고리즘 [LSRG15]	8-비트 AVR	2,165,239	2,617,459	686,367
Ring-LWE 기반 암호화 알고리즘 [ALSK15]	32-비트 ARM	1,232,000	1,452,000	328,000

표 4. 8-비트 AVR과 32-비트 ARM에서 Ring-LWE 기반 암호화 알고리즘 암/복호화 연산 결과(cycles)

격자 기반 전자서명 알고리즘으로는 BLISS가 가장 빠른 가장 빠른 것으로 알려져 있다. 표 5은 32MHz, 128 kB flashable program memory, 8 kB SRAM을 갖는 8-비트 ATxmega128A1 processor에서 128-비트 안전도의 BLISS-I을 구현한 결과이다.

암호화 알고리즘	서명생성	서명검증
BLISS-I [POG15]	10,537,981	2,814,118

표 5. 8-비트 AVR에서 BLISS-I의 서명생성/검증 결과(cycles)

표 6은 BLISS와 RSA, ECDSA의 서명길이 (signature size), 공개키 크기 (PK size), 비밀키 크기 (SK size)와 Intel Core 3.4 GHz에서 서명생성 시간(sign), 서명검증 시간(verify)을 구현한 결과를 비교한 표이다. [DDLL13]

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Implementation	Security	Signature Size	SK Size	PK Size	Sign (ms)	Sign/s	Verify (ms)	Verify/s
BLISS-0	⩽ 60 bits	3.3 kb	1.5 kb	3.3 kb	0.241	4k	0.017	59k
BLISS-I	128 bits	5.6 kb	2 kb	7 kb	0.124	8k	0.030	33k
BLISS-II	128 bits	5 kb	2 kb	7 kb	0.480	2k	0.030	33k
BLISS-III	160 bits	6 kb	3 kb	7 kb	0.203	5k	0.031	32k
BLISS-IV	192 bits	6.5 kb	3 kb	7 kb	0.375	2.5k	0.032	31k
RSA 1024	72-80 bits	1 kb	1 kb	1 kb	0.167	6k	0.004	91k
RSA 2048	103-112 bits	2 kb	2 kb	2 kb	1.180	0.8k	0.038	27k
RSA 4096	⩾ 128 bits	4 kb	4 kb	4 kb	8.660	0.1k	0.138	7.5k
ECDSA[1] 160	80 bits	0.32 kb	0.16 kb	0.16 kb	0.058	17k	0.205	5k
ECDSA 256	128 bits	0.5 kb	0.25 kb	0.25 kb	0.106	9.5k	0.384	2.5k
ECDSA 384	192 bits	0.75 kb	0.37 kb	0.37 kb	0.195	5k	0.853	1k

표 6. 여러 가지 안전도의 BLISS와 RSA, ECDSA를 구현한 결과 비교

3. 다변수 이차식 기반 공개키 암호알고리즘

다변수 이차식 기반 공개키 암호알고리즘은 다변수 이차식 시스템의 해를 찾는 문제(MQ-문제)가 어렵다는 NP-hard 문제를 이용한 공개키 암호알고리즘으로 1990년대부터 관련 암호화 알고리즘과 전자서명 알고리즘들이 많이 제안이 되었으나 오랜 검증 과정을 거치면서 모두 공격을 받아 안전하지 않다는 것이 밝혀졌다. 암호화 알고리즘은 거의 모두 깨졌으며, HFEv-의 변형과 Unbalanced Oil and Vinegar (UOV), Rainbow의 전자서명들만이 안전하게 남아있는 상황이다. 다변수 이차식 기반 공개키 암호알고리즘은 MQ-문제로만은 trapdoor를 숨길 수 없어 Isomorphism of Polynomials (IP) 문제를 이용하고 있는데 이런 MQ+IP 패러다임에 있는 다변수 이차식 기반 공개키 암호알고리즘들은 F_{31}, F_{2^8} 등 아주 작은 유한체의 연산을 요구하므로 속도가 매우 빠르다는 장점이 있어 리소스가 제한된 경량 기기에 적합한 암호로 평가받고 있다. 그러나, 키 길이가 다른 암호알고리즘 보다 길고, 안전성 증명이 이루어지지 않고 있다는 단점을 가지고 있다. 최근, Asiacrypt 2016에서 기존에 제안된 3-pass identification protocol을 Fiat-Shamir transform을 이용하여 안전성이 증명 가능한 다변수 이차식 기반 전자서명 MQDSS를 구현한 결과가 발표되었다. 이 MQDSS는 IP-문제를 이용하지 않고 MQ-문제의 random instance로의 안전성 귀착을 갖는다는 것을 보였으며, 키 길이도 현재 사용하고 있는 공개키 암호와 비슷한 정도로 짧지만, 전자서명의 길이가 128-비트 안전도에서 41kB를 육박하고 서명생성과 서명검증의 너무 비효율적이라 기존 MQ+IP 패러다임의 다변수 이차식 기반 공개키 암호알고리즘의 장점을 유지하지 못하고 있다.

표 7은 다변수 이차식 기반 전자서명 UOV와 Rainbow를 32MHz, 128 kB flashable program memory, 8 kB SRAM을 갖는 8-비트 ATxmega128A1 processor에서 구현한 결과를 나타낸 것이다[CHT12]. 표 8의 다변수 이차식 기반 전자서명 알고리즘의 상세는 다음과 같다.

- UOV: Oil-Vinegar 방법을 이용한 다변수 이차식 기반 전자서명

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

- 0/1 UOV: 특별한 ordering을 이용하여 UOV의 키 길이를 줄인 전자서명
- Rainbow: UOV에 층을 두어 효율성을 높인 전자서명
- enTTS: Rianbow의 특별한 경우로써 sparse 다항식을 이용한 전자서명

	Scheme	n	m	Key Size [Byte] private	public	System Parameter private	public	Clockcyles x 1000 sign	verify	Time[ms]@32MHz sign	verify	Code Size [Byte] sign	verify
	enTTS(5, 20, 28)	28	20	1351	8120	*	*	153	1,126	4.79	35.22	12890	827
	enTTS(5, 20, 28)[25]	28	20	1417	8680	*	*	568[1]	5,808[1]	17.75[2]	181.5[2]	-	-
2^{64}	uov(21, 28)	49	21	21462	25725	*	*	1,615	1,690	50.49	52.83	2188	466
	0/1 uov(21, 28)	49	21	12936	4851	8526	20874	1,577	1,395	49.29	43.60	2258	578
	rainbow(15, 10, 10)	35	20	9250	12600	*	*	848	1,010	26.51	31.58	4162	466
	enTTS(7, 28, 40)	40	28	2731	22960	*	*	332	2,558	10.37	79.95	24898	827
2^{80}	uov(28, 37)	65	28	49728	60060	*	*	3,637	3,911	113.66	122.23	2188	466
	0/1 uov(28, 37)	65	28	30044	11368	19684	48692	3,526	3,211	110.20	100.37	2258	578
	rainbow(18, 13, 14)	45	27	19682	27945	*	*	1,740	2,214	54.38	69.19	4162	466
	enTTS(9, 36, 52)	52	36	4591	49608	*	*	609	6,658	19.03	208.07	41232	827
2^{128}	uov(44, 59)	103	44	194700	235664	*	*	13,314	14,134	416.07	441.70	2188	466
	0/1 uov(44, 59)	103	44	116820	43560	77880	192104	12,782	13,569	399.43	424.04	2258	578
	rainbow(36, 21, 22)	79	43	97675	135880	*	*	8,227	9,216	257.11	288.01	4162	466
	enTTS(15, 60, 88)	88	60	13051	234960	*	*	2,142	3,0789	66.94	962.17	116698	827

표 7. 8-비트 AVR에서 다변수 이차식 기반 전자서명 UOV, 0/1 UOV, Rainbow, enTTS 키 길이와 서명생성, 서명검증 구현결과

4. 다양한 난제 기반 공개키 암호알고리즘 효율성 비교 분석

여기서는 리소스가 제한된 8-bit, 16-bit, 32-bit microprocessor에서 다양한 난제 기반 공개키 암호알고리즘의 효율성을 비교 분석한다. 표 8은 다양한 난제 기반 공개키 암호알고리즘의 8-bit, 16-bit, 32-bit platform에서 구현한 결과를 비교한 표이다. 표 8에 있는 전자서명 알고리즘과 platform 상세는 다음과 같음.

- NIST P-256: NIST 표준 타원곡선, Curve25519: Twisted Edwards curve
- SPHICS 256: 해쉬함수 기반 stateless 전자서명 알고리즘
- XMSS^{MT}: 해쉬함수 기반 stateful 전자서명 알고리즘
- BLISS-I: 격자기반 전자서명
- Rainbow: 다변수 이차식 기반 전자서명
- enTTS: Rianbow의 특별한 경우로써 sparse 다항식을 이용한 전자서명
- Curve 22519: Twisted Edwards curve인 Ed25519와 birationally equivalent한 타원곡선
- AX 128: ATxMega128a1, 32 MHz, 128 flash memory, 8 kB RAM
- AT 2560: ATmega2560, 16 MHz

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

비트	Platform	전자서명	서명생성	서명 검증	기반 문제
8-bit	AX 128	Rainbow(36,21,22)	8,227,000	9,216,000	MQ 문제, IP 문제, MinRank 문제
	AX 128	enTTS(15,60,88)	2,142,000	30,780,000	MQ 문제, IP 문제, MinRank 문제
	AX 128	BLISS-I	10,537,981	2,814,118	Ring-SIS 문제
	AT 2560	Curve 25519	13,900,397	-	ECDLP
	AT 2560	NIST P-256	17,520,000	-	ECDLP
16-bit	MSP 430X 16-bit H/W multiplier	NIST P-256	7,284,377	-	ECDLP
		Curve 25519	7,933,296	-	ECDLP
	MSP 430X 16-bit H/W multiplier	NIST P-256	5,321,776	-	ECDLP
		Curve 25519	5,301,792	-	ECDLP
32-bit	ARM Cortex M3	SPHINCS 256	729,942,616	17,707,814	collision resistant 2nd preimage resistant
		XMSS^{MT}	22,725,092	5,528,712	hash collision
	ARM Cortex M0	NIST P-256	2,762,000	-	ECDLP
		Curve 25519	3,589,850	-	ECDLP

표 8. 다양한 난제 기반 공개키 암호알고리즘 효율성 비교 분석

표 8에 나타난 기반문제의 상세는 다음과 같다.

- ECDLP(타원곡선 이산대수 문제): 유한체에서 정의된 타원곡선군 $E(F_q)$와 생성원 P와 $xP \in E(F_q)$가 주어져 있을 때, x를 구하는 문제
- MQ-문제: 다변수 이차식으로 구성된 시스템 $p_1(x)=p_2(x)=\dots p_n(x)=0$의 해를 찾는 문제, 여기서 $p_i(x)$는 유한체 F_q에서 정의된 $x=(x_1,\dots,x_m)$에 대한 이차식임
- IP-문제 (Isomorphism of Polynomials): 다항식 P, P'이 주어졌을 때, 두 다항식 사이의 isomorphism S, T, 즉 $P=T\circ P'\circ S$를 만족하는 S, T를 찾는 문제
- MinRank 문제: $F^{n\times n}$에서 정의된 행렬 $M_1,\dots,M_k$과 자연수 r에 대해, $\mathrm{Rank}(\sum_{i=1}^{k}\lambda_i M_i)\le r$을 만족하는 λ_i들을 찾는, 즉, M_1, ..., M_k의 일차결합을 찾는 문제
- Rin-SIS 문제: R이 ring이고, K가 $R_q^{n\times m}$에서의 distribution일 때, K를 따르는 랜덤한 $A\in R_q^{n\times m}$이 주어졌을 때, $Av=0$와 $\|v\|\le\beta$를 만족하는 $v\in R_q^m$를 찾는 문제, 여기서 R_q는 quotient ring $R/(qR)$임
- collision resistant: H(a) = H(b)를 만족하는 서로 다른 a와 b를 찾는 문제
- 2nd preimage resistant: x가 주어졌을 때, h(x) = h(x')를 만족하는 a second preimage x' ($\neq$ x)을 찾는 문제

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 2 절. 다변수 이차식 기반 공개키 암호알고리즘

1. 다변수 이차식 기반 암호화 알고리즘과 전자서명 알고리즘

다변수 이차식 기반 암호알고리즘(MQ-PKC)은 양자 알고리즘에 안전할 것으로 여겨지는 또 하나의 후보이다. 격자 기반 암호시스템처럼, Shor의 양자 알고리즘 발표[S97] 전후인 90년대부터 활발하게 연구가 이루어졌다. 하지만 격자 기반 암호시스템과는 달리, MQ-PKC는 안전한 암호화 알고리즘이 거의 없으며, 대부분은 서명 알고리즘이다. 여기에서는 MQ-PKC를 암호화 알고리즘과 서명 알고리즘으로 나누어 살펴보기로 한다.

가. 기반 문제

다변수 이차식의 시스템을 푸는 문제는 일반적으로 NP-hard로 알려져 있다. 즉, 기반 문제 자체는 충분히 어려운 문제이다. 하지만 실제 안전한 공개키 암호시스템을 만드는 데에는 충분하지 않다. 여기서는 다변수 이차식 기반 암호알고리즘에서 쓰이는 두 개의 추가 문제들에 대해 살펴보기로 한다.

(1) Multivariate Quadratic Problem (MQ 문제)

MQ 문제는 유한체위에서 정의된 다변수 이차 연립 방정식의 해를 구하는 것이다. 구체적으로 n개의 변수 $x_1,...,x_n$에 대한 m개의 이차 방정식 $P=(p^{(1)},...,p^{(m)})$이 주어져 있을 때, $p^{(1)}(x_1',...,x_n')=...=p^{(m)}(x_1',...,x_n')=0$을 만족하는 해 $(x_1',...,x_n')$를 구하는 문제이다. 이 문제는 $n\approx m$인 경우 유한체 F_2위에서 NP-complete임이 증명이 되었다. 그러나 $n\gg m$(highly under-determined)이거나 $n\ll m$(highly over-determined)인 경우, 다항식시간에 푸는 방법이 알려져 있다. 일반적으로 MQ 문제를 해결하는 알고리즘으로 gröbner basis를 계산하여 푸는 방법이 현재까지는 가장 빠른 것으로 알려져 있다.

(2) Isomorphism of Polynomials (IP 문제)

유한 체에서의 방정식에 기반을 둔 안전한 암호알고리즘을 만들기 위해서는 IP 문제(Isomorphism of Polynomials)[P96]가 또한 매우 중요하다. 이 문제는, 주어진 다항식 벡터들 P와 P'에 대해, $P=T\circ P'\circ S$이 되는 아핀 변환 $S\in Aff^{-1}(\mathbb{F}^n)$와 $T\in Aff^{-1}(\mathbb{F}^m)$를 찾는 문제이다. 보통 (S,P',T)가 비밀키, P가 공개키 역할을 한다. 그렇기 때문에 만약 IP 문제가 쉽다면 이러한 스킴들의 안전성은 심각하게 위협을 받는다. 따라서 이러한 스킴을 만들 때에는 IP 문제가 어렵다는 가정이 필요하다. 하지만 문제는, P'의 구조에 따라 그에 대응되는 IP 문제가 쉬워지고, 그에 따라 시스템 전체가 위험해지는 경우가 많

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

다는 것이다. 많은 MQ-PKC들이 제시되었지만 대다수가 깨진 이유이며, 근본적인 해결은 아직 되지 않고 있는 상황이다.

(3) MinRank 문제

자연수 k에 대해, $(M_1, ..., M_k)$를 $\mathbb{F}^{n\times n}$의 행렬들이라 하자. 자연수 r에 대해, MinRank 문제는, rank가 r 이하인 선형 결합을 찾는 것이다. 즉, $Rank(\sum_{i=1}^{k}\lambda_i M_i) \le r$인 벡터 $\lambda \in \mathbb{F}^k$를 찾는 문제이다. 유한 체에서 이 문제는 NP-complete임이 증명되어 있다[BFS97]. 하지만 특정한 경우, 예를 들어 rank r이 굉장히 작거나, 또는 행렬 $M_1, ..., M_k$ 중 가장 큰 rank $R \in \mathbb{N}$이 n에 아주 가까운 경우, 이 문제는 쉬워진다. [GC00]에서 이 두 가지 경우에 대해 복잡도가 각각 $O(q^r)$과 $O(q^{n-R})$인 알고리즘을 제시하였다. 이 경우들에 대해 더 효율적인 알고리즘이 있는지, 혹은 일반적인 MinRank 문제에 대한 좋은 알고리즘이 있는지는 아직 정확히 알려지지 않고 있다. 만약 그러한 알고리즘들이 있다면, 다변수 이차식 기반 암호시스템들의 안전성에 심각한 문제가 생기게 된다.

나. 다변수 이차식 기반 전자서명 알고리즘

다변수 이차식 기반 전자서명 알고리즘은 작은 유한체를 이용하는 Single-field 유형과 기본 유한체와 상대적으로 크기가 큰 확장체를 이용하는 Mixed-field 유형으로 나눌 수 있다. Single-field 유형에 속하는 전자서명은 UOV, Rainbow, TTS, MQQ-SIG 등이 있으며, Mixed-field 유형에 속하는 것은 HFEv-의 변형들이 있다.

(1) TTS

TTS는 대만의 연구진들이 개발하였으며, 2002년 대만 학술회의에서 처음 소개되었지만, 국제적으로는 2005년의 논문[YC05]을 완성본으로 보고 있다. TTS 전자서명은 다음과 같다.

■ TTS 전자서명

TTS는 다음과 같은 구조를 사용한다.

$$V = \phi_3 \circ \phi_2 \circ \phi_1 : K^n \to K^m,\ \phi_1 : \boldsymbol{w} \mapsto \boldsymbol{x} = M_1\boldsymbol{w} + \boldsymbol{c_1},\ \phi_3 : \boldsymbol{y} \mapsto \boldsymbol{z} = M_3\boldsymbol{y} + \boldsymbol{c_3}$$

각각 K^n과 K^m에서의 아핀 맵이고 보통 역변환 가능하다. ϕ_2를 중간 맵이라 하고, 각각의 y_j를 x_i들로 표현하는 방정식을 중간 방정식이라 한다. $q = |K|$, r은 중간 방정식들의 선형 결합에서 가장 작은 rank라 놓는다. 일반성을 잃지 않고, 첫 번째 중간 방정식의

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

rank라고 할 수 있다. 또한, 일반성을 잃지 않고, 중간 방정식들의 cross-term들에서 가장 적게 나타나는 변수를 x_{n-1}이라 하고, u회 나타난다고 하자. 저자들은 2^{80}의 안전성을 가지기 위한 최소 파라미터인 TTS(20, 28)에 대하여 서술하였다. 파라미터 p_{ij}에 $K = GF(2^8)$의 원소들을 임의로 정해준다. 임의의 정칙행렬 $M_1 \in K^{28\times 28}$과 $M_3 \in K^{20\times 20}$를 만든다. 보통 LU decomposition을 이용한다. 임의의 벡터 $\boldsymbol{c_1} \in K^{28}$도 고른다. $V = \phi_3 \circ \phi_2 \circ \phi_1$으로 놓고, $\boldsymbol{c_3} \in K^{20}$을 V에 상수 부분이 없도록 잡는다. 2^{80} 3DES 블록들의 복잡도를 가지기 위해서는 다음과 같은 조건들이 필요하다.

○ 해시가 160비트가 되거나, $m \geq 20$(생일 공격을 막기 위해), $n \geq m$
○ XL/F_5 공격을 막기 위해 $m \geq 20$ ($q = 2^7$이면 $m \geq 22$가 필요)
○ $r > 8$, 즉, 각 방정식마다 독립적인 cross-term이 적어도 5개 이상 있어야 한다. 특별한 공격을 막기 위해 6이나 7개가 필요할 수 있다.
○ 키 길이나 실행 시간 등을 고려하면 n이 너무 크지 않은 것이 좋다.
○ $u \geq 9$, 즉, 모든 변수들은 적어도 9개 이상의 방정식에는 나타나야 한다.

이에 따라 중간 맵 ϕ_2를 다음과 같이 정할 수 있다.

$$y_i = x_i + \sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j \bmod 9)}, \quad i = 8, \ldots, 16;$$

$$y_{17} = x_{17} + p_{17,1}x_1x_6 + p_{17,2}x_2x_5 + p_{17,3}x_3x_4 + p_{17,4}x_9x_{16} + p_{17,5}x_{10}x_{15} + p_{17,6}x_{11}x_{14} + p_{17,7}x_{12}x_{13}$$

$$y_{18} = x_{18} + p_{18,1}x_2x_7 + p_{18,2}x_3x_6 + p_{18,3}x_4x_5 + p_{18,4}x_{10}x_{17} + p_{18,5}x_{11}x_{16} + p_{18,6}x_{12}x_{15} + p_{18,7}x_{13}x_{14}$$

$$y_i = x_i + p_{i,0}x_{i-11}x_{i-9} + \sum_{j=19}^{i-1} p_{i,j-18}x_{2(i-j)-(i \bmod 2)}x_j + p_{i,i-18}x_0x_i + \sum_{j=i+1}^{27} p_{i,j-18}x_{i-j+19}x_j,$$
$$i = 19, \ldots, 27$$

☞공개키
V의 이차항과 일차항의 계수들 (8680바이트)

☞비밀키
M_1^{-1}, M_3^{-1}, $\boldsymbol{c_1}$, $\boldsymbol{c_3}$, 파라미터 p_{ij} 값들 (1399바이트)

☞서명 생성
-메시지 M에 대해, 해시 값 $\boldsymbol{z} = H(M) \in K^{20}$을 계산하고, $\boldsymbol{y} = M_3^{-1}(\boldsymbol{z} - \boldsymbol{c_3})$를 계산한 후, 가능한 $\boldsymbol{x} \in \phi_2^{-1}(\boldsymbol{y})$ 값을 계산한다. 계산 방법은 다음과 같다. 먼저 $x_1, \ldots, x_7$을 정해주고, x_8에서 x_{16}에 대해 처음 아홉 개의 방정식을 풀어본다. 실패하면 다시 새로 정해서 풀어본다. 이 시스템이 풀릴 확률은 대략 255/256이다. 어떤 x_1에서 x_6까지든지 첫

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

시스템의 행렬식은 x_1에 대한 9차 다항식이므로 최대 9가지의 x_1에 대해 첫 시스템이 degenerate하게 된다. 그러므로 시스템을 풀 확률은 적어도 247/256이다.

-다음 두 방정식 y_{17}과 y_{18}을 이용해 차례로 x_{17}과 x_{18}을 구한다. 임의의 x_0를 잡고 x_{19}부터 x_{27}의 두 번째 시스템을 풀어본다. 역시 두 번째 시스템의 행렬식을 0으로 만드는 x_0가 최대 9개이므로, 시스템을 푸는 데 실패하면 다른 x_0를 잡아 해를 구한다.

-이렇게 해서 계산한 $\boldsymbol{w}= M_1^{-1}(\boldsymbol{x}-\boldsymbol{c_1})$이 서명 값이 된다. $(M, \boldsymbol{w})$를 공개한다.

☞서명 검증

$\boldsymbol{z}= H(M)$을 계산하여 $V(\boldsymbol{w})$와 일치하는지 확인한다.

(2) UOV(Unbalanced Oil and Vinegar)

1997년 Patarin이 Oil and Vinegar라는 서명 알고리즘을 발표하였다[P97]. 하지만 취약점이 금방 드러났으며[KS98], 이에 따라 이 알고리즘을 일반화한 UOV(Unbalanced Oil and Vinegar) 알고리즘이 제안되었다[KPG99]. UOV는 다음과 같다.

☞유한체, 변수 선택

$K=\mathbb{F}_q$: 작은 유한 체

o, v : 정수, $n=o+v$

$V:=\{1,...,v\}$, $O:=\{v+1,...,n\}$: vinegar 미지수들과 oil 미지수들의 인덱스 집합

☞비밀키 (F, S)

$F:=(f^{(1)},...,f^{(o)})$는 다음과 같이 정의한다.

$$f^{(k)}(u_1,...,u_n) := \sum_{i\in V,\, j\in O} f_{ij}^{(k)} u_i u_j + \sum_{i,\, j\in V,\, i\le j} f_{ij}^{(k)} u_i u_j,\ (k=1,...,o)$$

전단사 아핀 함수 $S: K^{o+v}\rightarrow K^{o+v}$

☞공개키

중심 맵 F에 선형 맵 $S: \mathbb{F}_q^n\rightarrow\mathbb{F}_q^n$을 합성하여 F를 숨긴다. 즉, 공개 키는 $P:=F\circ S$이다. 선형 맵 S가 정해지면 이 식을 통해 다음과 같은 P와 F의 이차 단항식 계수들에 대한 관계식을 얻을 수 있다.

$$p_{ij}^{(k)} = \sum_{r=1}^{n}\sum_{s=r}^{n} \alpha_{ij}^{rs} \cdot f_{rs}^{(k)}$$

위 식에서 $p_{ij}^{(k)}$와 $f_{ij}^{(k)}$는 각각 P와 F의 k번째 요소에서 $x_i x_j$의 계수를 뜻하며, α_{ij}^{rs}는 다음과 같이 주어진다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$\alpha_{ij}^{rs} = \begin{cases} s_{ri} \bullet s_{si} & (i=j) \\ s_{ri} \bullet s_{sj} + s_{rj} \bullet s_{si} & otherwise \end{cases}$$

여기에서 $s_{ij} \in \mathbb{F}_q$는 선형 맵 S의 계수들을 의미한다.

☞서명 생성

- 메시지 M이 주어졌을 때, $h(M) = y = (y_1, ..., y_o) \in K^o$을 계산한다.
- o개의 미지수 $a_{v+1}, ..., a_n$와 v개의 미지수 $a_1, ..., a_v$를 다음과 같이 찾아낸다. 먼저 v개의 vinegar 미지수 a_i'을 임의로 선택하여 미지수 $a_1, ..., a_v$에 대입하면 o개의 미지수를 갖는 o개의 일차식의 시스템을 얻게 된다. 이 시스템에서 가우스 소거법을 이용해 해 $a_{v+1}', ..., a_n'$을 구하게 된다. 만약 해가 존재하지 않는다면 vinegar 미지수들을 새로 골라 해를 찾을 때까지 반복한다. 몇 번만 하면 적어도 한 개의 해를 얻을 확률이 매우 높다.
- $A = (a_1', ..., a_{o+v}') \in K^{o+v}$일 때, $(b_1, ..., b_{v+o}) = S^{-1}(A)$를 계산하여 이것을 서명 값으로 한다.

☞서명 검증

$P(b) = h(M)$이 성립하는지 확인한다.

(3) Rainbow

Rainbow 서명 알고리즘은 UOV 서명 알고리즘을 일반화한 서명 알고리즘으로 2005년 Ding과 Schmidt에 의해 제안되었다[DS05]. Rainbow 서명 알고리즘에서는 기존 1개의 layer만을 사용한 UOV에 비해 layer를 여러 개로 늘려 키의 길이와 서명 생성 속도를 개선하였다. Rainbow는 $P = F \circ T$의 구조를 사용한 UOV와 달리, 역 연산이 가능한 affine map S와 T 두 개를 적용한 $P = S \circ F \circ T$의 구조를 사용한다. Central map의 역연산을 하는 법을 알아보기 전에 Rainbow에서 필요한 각 파라미터들을 다음과 같이 정의하기로 한다.

▪ Rainbow의 파라미터

- ○ $K = \mathbb{F}_q$: 이 스킴이 기반으로 하는 유한체이다.
- ○ L : Layer의 개수
- ○ o_l : 각 $l(\leq L)$번째 layer에서의 다항식의 개수(양의 정수) = l번째 layer에서의 vinegar 변수의 개수
- ○ $m = o_1 + ... + o_L$: 다항식의 총 개수
- ○ v_l : 각 l번째 layer에서의 vinegar 변수의 개수. $l > 1$이면, $v_l = v_{l-1} + o_{l-1}$이 성립한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

○ $n = m + v_1$: 변수의 총 개수

○ $m_l = o_1 + ... + o_l$: l번째 layer까지의 다항식의 개수

○ $n_l = m_l + v_1$: l번째 layer까지의 변수의 개수

Central map인 $F: K^n \to K^m = (f^{(1)}, ..., f^{(m)})$의 각 다항식 $f^{(k)}$를 정의하기 위해 위의 파라미터를 적용하자. 만약 $m_{l-1} < k \leq m_l$이면, 즉 $f^{(k)}$가 l번째 layer의 다항식이라면,

$$f^{(k)}(x_1, x_2, ..., x_n) = \sum_{i \leq j \leq v_l} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \leq v_l < j \leq n_l} \beta_{ij}^{(k)} x_i x_j + \sum_{0 < i \leq n_l} \gamma^{(k)} x_i + \eta^{(k)}$$

의 형태가 된다. UOV에서처럼 (l번째 layer에서의 oil 변수)×(l번째 layer에서의 oil 변수)의 계수는 모두 0이다.

▪ 주어진 $y = (y_1, ..., y_m) \in K^m$에 대하여 $F(x_1, ..., x_n) = y$가 되는 $(x_1, ..., x_n) \in K^n$ 찾기

○ 1번째 layer

- 1번째 layer에서의 vinegar 변수들($x_1, ..., x_{v_1} \in K$)을 임의의 값으로 설정하여 각 $f^{(k)}(x_1, ..., x_n) = y_k$ $(1 \leq k \leq o_1)$에 대입한다.
- (1번째 layer의 oil 변수)×(1번째 layer의 oil 변수)의 계수와 $x_i x_j (i, j > n_1)$의 계수가 모두 0이므로 위의 식에는 2차항이 없다. 따라서 이는 o_1개의 식에 변수가 o_1개인 1차 연립방정식 시스템이 되므로, Gauss 소거법을 사용하여 $x_{v_1+1}, ..., x_{v_1+o_1} (= x_{n_1} = x_{v_2})$을 찾을 수 있다.

○ 2번째 layer

- 1번째 layer까지 찾은 변수들($x_1, ..., x_{n_1}$)이 2번째 layer의 vinegar 변수들이 된다.
- (2번째 layer의 oil 변수)×(2번째 layer의 oil 변수)의 계수와 $x_i x_j (i, j > n_2)$의 계수가 모두 0이므로 위의 식에는 2차항이 없다. 따라서 이는 o_2개의 식에 변수가 o_2개인 1차 연립방정식 시스템이 되므로, Gauss 소거법을 사용하여 $x_{v_2+1}, ..., x_{v_2+o_2} (= x_{n_2} = x_{v_3})$을 찾을 수 있다.

–

○ l번째 layer

- $l-1$번째 layer까지 찾은 변수들($x_1, ..., x_{n_{l-1}}$)이 l번째 layer의 vinegar 변수들이 된다.
- (l번째 layer의 oil 변수)×(l번째 layer의 oil 변수)의 계수와 $x_i x_j (i, j > n_l)$의 계수가 모두 0이므로 위의 식에는 2차항이 없다. 따라서 이는 o_l개의 식에 변수가 o_l개인 1차 연립방정식 시스템이 되므로, Gauss 소거법을 사용하여 $x_{v_l+1}, ..., x_{v_l+o_l} (= x_{n_l} = x_{v_{l+1}})$을 찾을 수 있다.

–

○ 만약 각각의 layer에서의 연립 1차방정식 시스템에서 해를 찾을 수 없었다면 첫 번째 단계로 되돌아가서 다시 vinegar 변수들을 선택한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

layer의 개수가 너무 많지 않다면 위의 방법을 사용하여 F의 역연산을 수행할 수 있을 확률이 매우 높다고 알려져 있다. F는 특수한 형태이지만, affine invertible map인 S와 T로 인해 변수가 섞이게 된다. Rainbow 서명 알고리즘의 상세한 사항은 다음과 같다.

- 키 생성
  - 위에서 소개한대로 파라미터를 설정한다.
  - $F: K^n \to K^m = (f^{(1)},...,f^{(m)})$를 위에서 설명한 형태를 갖도록 각 $\alpha_{ij}^{(k)}, \beta_{ij}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in K$를 임의로 선택한다.
  - 역연산이 가능한 affine maps $S: K^m \to K^m$, $T: K^n \to K^n$를 임의로 선택한다.
  - $P: K^n \to K^m = S \circ F \circ T$를 계산한다.
  - P가 공개키이고, S와 F와 T가 비밀키이다.

- 서명 생성 : 메시지 $\overline{M} = (\overline{M_1}, \ldots, \overline{M_m}) \in K^m$에 대한 서명값 $\overline{S} = (\overline{S_1}, \ldots, \overline{S_n}) \in K^n$ 계산
  - S의 역 연산을 사용하여 $S(y) = S(y_1, \ldots, y_m) = (\overline{M_1}, \ldots, \overline{M_m})$이 되는 $y = (y_1, \ldots, y_m) \in K^m$를 계산한다.
  - 위에서 소개한 방법을 이용하여 $F(x_1,...,x_n) = y$이 되는 $(x_1,...,x_n) \in K^n$을 찾아낸다.
  - T의 역 연산을 사용하여 $T(s) = T(\overline{S_1}, \ldots, \overline{S_n}) = (x_1, \ldots, x_n)$이 되는 $\overline{S} = (\overline{S_1}, \ldots, \overline{S_n}) \in K^n$를 계산한다.

- 서명 검증 : $P(\overline{S}) = \overline{M}$인지 확인한다.

Rainbow는 복층구조와 2개의 affine map을 사용하는 이유로 인해 UOV보다 훨씬 작은 키 길이를 가지고도 비슷한 안전성을 가지도록 설계할 수 있다. Rainbow의 설계자들이 [DS05]에서 소개한 예에 따르면, UOV보다 약 1/10의 키 길이를 갖더라도 비슷한 안전성을 갖도록 설계할 수 있다.

(4) MQQ-SIG

위에서 알아본 대표적인 스킴들 이외에도 많은 다변수 이차식 기반 암호알고리즘들이 제안되고 있다. 여기에서는 그 중 가장 주목을 받고 있는 MQQ 서명 스킴[GOJP12]에 대하여 알아보기로 한다. MQQ는 Multivariate Quadratic Quasigroups의 약자이다. Quasigroup은 다음과 같이 정의된다.

정의 1. Quasigroup $(Q, *)$는 다음 법칙을 만족하는 groupoid이다.

$$(\forall u, v \in Q)\,(\exists! x, y \in Q)\; u*x = v \;\; y*u = v.$$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

위의 정의는, 나눗셈 혹은 소거가 항상 가능하다는 것을 의미한다. 즉, $x*y=x*z$이면 $y=z$가 되고, $y*x=z*x$이면 $y=z$가 된다. 또한 방정식 $a*x=b$, $y*a=b$가 각각의 $a,b\in Q$에 대해 유일한 해 x,y를 가지게 된다. 이제부터는 차수가 2^d, 즉 $|Q|=2^d$인 quasigroup을 다룰 것이다. MQ-PKC를 정의하기 위해 다음 결과가 필요하다.

보조정리 1. 차수 2^d인 모든 quasigroup $(Q,*)$와 각 bijection $Q\rightarrow\{0,1,...,2^d-1\}$에 대해, a개의 유일하게 결정되는 vector valued Boolean 함수 $*_{vv}$와 d개의 유일하게 결정되는 $2d$-ary Boolean 함수 $f_1,f_2,...,f_d$가 존재한다. 여기에서 각 $a,b,c\in Q$에 대해 연산 $a*b=c$는 다음과 같이 나타내어진다.

$$*_{vv}(x_1,...,x_d,y_1,...,y_d)=(f_1(x_1,...,x_d,y_1,...,y_d),...,f_d(x_1,...,x_d,y_1,...,y_d))$$

정의 2. 차수 2^d인 Quasigroup $(Q,*)$에 대해, $0\le k<d$일 때, 다항식 f_i 중 정확히 $d-k$개가 2차식이고 k개가 1차식이면, 그것을 타입 $Quad_{d-k}Lin_k$인 Multivariate Quadratic Quasigroup(MQQ)이라 부른다.

중간 전단사 다변수 이차 맵 $P':\{0,1\}^n\rightarrow\{0,1\}^n$을 다음과 같은 방법으로 정의한다. n개의 변수로 이루어진 n개의 일차 Boolean 함수 벡터 $\vec{x}=(f_1,...,f_n)$이 주어지고, multivariate quadratic quasigroup *가 미리 정의되어 있으며, $k\in\{5,6,7,8\}$일 때 $n=32\times k$가 주어져 있다. 이 벡터 $\vec{x}$를 X_i가 차수 8인 벡터일 때 스트링 $\vec{x}=X_1...X_{\frac{n}{8}}$으로 나타낸다. 그리고 $Y_1=X_1$, $j=2,4,...$일 때 $Y_{j+1}=X_j*X_{j+1}$, $j=3,5,...$일 때 $Y_{j+1}=X_{j+1}*X_j$와 같이 계산하여 $\vec{y}=Y_1...Y_{\frac{n}{8}}$이라 한다. 이 때 $\vec{y}$가 중간 맵 P'이 된다. 차수 2^d인 MQQ *는 간단하게 다음과 같은 식으로 나타낼 수 있다.

$$\vec{x}*\vec{y}\equiv \boldsymbol{B}\cdot \boldsymbol{U}(\vec{x})\cdot \boldsymbol{A_2}\cdot \vec{y}+\boldsymbol{B}\cdot \boldsymbol{A_1}\cdot \vec{x}+\vec{c}$$

여기서 $\vec{x}=(x_1,...,x_d)$, $\vec{y}=(y_1,...,y_d)$이고, $\boldsymbol{A_1},\boldsymbol{A_2},\boldsymbol{B}$는 GF(2)에서 크기 $d\times d$인 nonsingular 행렬, $\vec{c}$는 GF(2)의 원소들로 이루어진 임의의 d차 벡터이며, 모두 uniformly random하게 생성한다. $\boldsymbol{U}(\vec{x})$는 upper triangular 행렬로, 모든 diagonal 원소는 1이고 diagonal 위쪽은 $\vec{x}=(x_1,...,x_d)$의 변수들로 이루어진 일차 표현들이다. $\boldsymbol{U_i}$는 행 $\{1,...,i\}$에서 diagonal보다 위쪽에 있는 원소들을 제외한 모든 원소가 0인 행렬일 때, 다음과 같은 식으로 계산할 수 있다. 원소가 0이냐 1이냐 역시 uniformly random하게 정한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$\boldsymbol{U}(\vec{x}) = I + \sum_{i=1}^{d-1} \boldsymbol{U_i} \bullet \boldsymbol{A_1} \bullet \vec{x}$$

또한, 다음 두 가지 조건도 만족해야 한다.

$$\forall i \in \{1, ..., d\}, Rank(\boldsymbol{B_{f_i}}) \geq 2d-4$$
$$\exists j \in \{1, ..., d\}, Rank(\boldsymbol{B_{f_j}}) = 2d-2$$

여기서 $\boldsymbol{B_{f_i}}$는 $2d \times 2d$ 행렬로, f_i로부터 다음과 같이 정의된다.

$$\boldsymbol{B_{f_i}} = [b_{j,k}], b_{j,d+k} = b_{d+k,j} = 1, \text{ iff } x_j y_k \text{가 } f_i \text{의 항.}$$

Nonsingular Boolean 행렬도 특별한 방법으로 만들어야 한다. 임의로 생성된 $n \times n$ 크기의 nonsingular Boolean 행렬을 저장하는 데에 n^2 비트가 필요하게 된다. 여기서는 $\boldsymbol{S^{-1}}$를 서명 과정에 사용하게 되므로 이 행렬을 저장한다. 80비트에서 128비트 안전도를 위해 $n = 160, 192, 224, 256$을 쓰게 되는데, 이 때 저장 공간은 3.125에서 8 KB가 필요하게 된다. MQ 스킴에서 키 크기를 줄이는 데에 circulant 행렬을 사용하는 경우들이 있는데, 여기서는 한 개가 아니라 두 개를 사용하게 된다. 여기서 $n \times n$ circulant 행렬은 첫 번째 행부터 각 행을 하나씩 rotate 해가며 만드는 행렬을 의미한다. 선형 아핀 변환에 대해 비밀 정보를 압축하는 데에 다음과 같은 nonsingular 행렬 $\boldsymbol{S}$가 필요하다.

$$\boldsymbol{S^{-1}} = \bigoplus_{i=0}^{\frac{n}{16}} I_{\sigma_i^0} \oplus \bigoplus_{i=0}^{\frac{n}{16}+3} I_{\sigma_i^1}$$

여기서 $I_{\sigma_i^0}, i = \left\{0, 1, 2, ..., \frac{n}{16}\right\}$, $I_{\sigma_i^1}, i = \left\{0, 1, 2, ..., \frac{n}{16}+1\right\}$은 크기 n인 치환행렬이고, 연산 $\oplus$는 치환행렬 원소들의 비트별 XOR, 그리고 σ_i^0와 σ_i^1은 n개 원소에 대한 permutation으로 다음과 같이 정의된다.

σ_0^0 : $\{1, 2, ..., n\}$에 대한 임의의 permutation

$$\sigma_i^0 = RotateLeft(\sigma_{i-1}^0, 8), \ i = 1, ..., \frac{n}{16}$$

σ_0^1 : $\{1, 2, ..., n\}$에 대한 임의의 permutation

$$\sigma_i^1 = RotateLeft(\sigma_{i-1}^1, 8), \ i = 1, ..., \frac{n}{16}+1$$

σ_0^0와 σ_0^1는 위의 $\boldsymbol{S^{-1}}$가 nonsingular 행렬이 되도록(그리고 $\boldsymbol{S} = (\boldsymbol{S^{-1}})^{-1}$가 되도록) 정한다. $\boldsymbol{S}$로부터 다음과 같은 아핀 변환을 얻는다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$\boldsymbol{S}'(\vec{x}) = \boldsymbol{S} \bullet \vec{x} + \vec{v}$$

여기서 $\vec{v} = (v_1, v_2, ..., v_n)$는 n차 Boolean 벡터로, permutation $\sigma_0^1 = (s_1, s_2, ..., s_n)$으로부터 다음과 같이 정의된다.

$$v_i = \left(\left(\frac{\left(\left(s_{1+\left\lfloor \frac{i-1}{8} \right\rfloor} \right) \bmod 16 \right) \times 16}{2^{(8-i) \bmod 8}} \right) + \left(\frac{s_{65+\left\lfloor \frac{i-1}{8} \right\rfloor}}{2^{(8-i) \bmod 8}} \right) \right) \bmod 2$$

위 식은 매우 복잡해 보이지만 소프트웨어와 하드웨어에서 매우 빨리 계산할 수 있다. MQQ-SIG 알고리즘은 다음과 같다.

■ MQQ-SIG

☞공개키

$n = 32 \times k$, $k = \{5, 6, 7, 8\}$일 때, 위와 같이 MQQ *와 nonsingular $n \times n$ Boolean 행렬 $\boldsymbol{S}$, 아핀 변환 $\boldsymbol{S}'$을 만든다. $\vec{x} = (x_1, ..., x_n)$일 때 $\vec{y} = \boldsymbol{S}(P'(\boldsymbol{S}'(\vec{x})))$를 계산한다. 이 때, $i = 1 + \frac{n}{2}, ..., n$일 때 $\frac{n}{2}$개의 다변수 이차방정식 $P_i(x_1, ..., x_n)$으로 주어지는 $\vec{y}$가 공개키이다.

☞비밀키

위 과정에서, $(\sigma_0^0, \sigma_0^1, *)$가 비밀키

☞서명 생성

$D(\vec{y})$를 역연산 $\boldsymbol{S}^{-1}, P'^{-1}, \boldsymbol{S}'^{-1}$의 합성이라 하자. 즉, $D(\vec{y}) \equiv \boldsymbol{S}^{-1}(P'^{-1}(\boldsymbol{S}'^{-1}(\vec{y})))$. 또한, $E(\vec{x})$는 벡터 $\vec{x}$를 공개 다항식들 $P_i(x_1, ..., x_n)$, $i = 1 + \frac{n}{2}, ..., n$로 보낸 값이라 하자. 서명할 문서 M과 해시함수 $Hash()$에 대해, $h = h_0 \| h_1 \leftarrow Hash(M)$을 계산한다. 해시함수의 결과는 n비트이고 h_0와 h_1은 각각 $\frac{n}{2}$비트라고 가정한다. Uniformly random하게 $\frac{n}{2}$비트 값 r_0와 r_1을 정하여 $\vec{y_0} = r_0 \| h_0$, $\vec{y_1} = r_1 \| h_1$으로 놓고, $\vec{x_0} = D(\vec{y_0})$, $\vec{x_1} = D(\vec{y_1})$을 계산한다. 이 때 M에 대한 서명 값은 $(\vec{x_0}, \vec{x_1})$이 된다.

☞ 서명 검증

$h = h_0 \| h_1 \leftarrow Hash(M)$을 계산하고, $\vec{z_0} = E(\vec{x_0})$, $\vec{z_1} = E(\vec{x_1})$을 계산한다. 이 때 $\vec{z_0} = h_0$이고 $\vec{z_1} = h_1$이 되면 TRUE, 아니면 FALSE를 리턴

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

(5) Cubic UOV

2015년 Inscrypt에서 Nie 등은 CUOV [NLXL15]라는 UOV [KPG99]에 기반한 새로운 다변수 이차식 기반 전자서명을 발표하였다. 그러나 2016년에 Hashimoto [H16]에 의해서 깨졌고, 2016년 ICISC에서 Duong 등에 의해서 공격에 대한 취약성을 보완한 새로운 버전이 발표되었다 [DPWT16]. 이 절에서는 이러한 UOV를 변형하여 새로운 다변수 이차식 기반 전자서명을 개발하려는 연구결과들에 대해서 살펴보겠다.

(가) CUOV 전자서명

2016년 Nie 등은 UOV 전자서명을 변형하여 새로운 이차식 기반 전자서명(MQ 전자서명)을 제안하였고, 이것을 Cubic Unbalanced Oil and Vinegar (CUOV)라고 이름을 붙였다. 이 스킴은 다음과 같다.

F_q를 원소의 개수가 q개인 유한체라 하고, 자연수 o,v에 대해서 $n=o+v$를 변수의 개수, o를 식의 개수라고 하자. 그러면 CUOV는 $\mathrm{F}: F_q^n \to F_q^o$인 함수 F와 $S: F_q^n \to F_q^n$인 가역인 아핀변환 S에 대해서 $P=\mathrm{F}\circ S$를 공개키로 함수 F와 아핀변환 S를 비밀키로 가지는 전자서명이다. 구체적으로 함수 F는 다음과 같이 만들어 진다.

키생성: 두 함수 $\overline{F}$와 $\hat{F}\times id_v$에 대해서 $\mathrm{F}=\overline{F}\circ(\hat{F}\times id_v)$로 정의된다. 먼저 $\hat{F}=(\hat{f}^{(1)},...,\hat{f}^{(o)}): F_q^n \to F_q^o$는 다음과 같이 정의된다.

$$\begin{cases} \hat{f}^{(1)} = \sum_{i=1}^{o}\sum_{j=o+1}^{n} a_{ij}^{(1)} y_i y_j + \sum_{i=o+1}^{n}\sum_{j=i}^{n} a_{ij}^{(1)} y_i y_j + \sum_{i=1}^{n} b_i^{(1)} y_i + c^{(1)} \\ \hat{f}^{(2)} = \sum_{i=1}^{n} b_i^{(2)} y_i + c^{(2)} \\ \cdots \\ \hat{f}^{(o)} = \sum_{i=1}^{n} b_i^{(o)} y_i + c^{(o)} \end{cases}$$

여기서 계수는 모두 F_q에서 임의로 선택된 원소이다. 그리고

$$\begin{aligned} \hat{F}\times id_v : F_q^n &\to F_q^n \\ (y_1,...,y_o,y_{o+1},...,y_n) &\mapsto (\hat{f}^{(1)},...,\hat{f}^{(o)},y_{o+1},...,y_n) \end{aligned}$$

로 정의된다. 함수 $\overline{F}$는 다음과 같이 정의된다.

$$\begin{aligned} \overline{F}: F_q^o \times F_q^v &\to F_q^o \\ (x_1,...,x_o,y_{o+1},...,y_n) &\mapsto (\overline{f}^{(1)},...,\overline{f}^{(o)}) \end{aligned}$$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$\overline{f}^{(i)}$는 다음과 같다.

$$\begin{cases} \overline{f}^{(1)} = r_1 \cdot (x_1 + x_1 x_2) + g_1(y_{o+1},...,y_n), \\ \overline{f}^{(2)} = r_2 \cdot x_1 x_2 + g_2(y_{o+1},...,y_n), \\ \overline{f}^{(3)} = r_3 \cdot (x_1 + x_2) x_3 + g_3(y_{o+1},...,y_n), \\ \quad ... \\ \overline{f}^{(o)} = r_o \cdot (x_{o-2} + x_{o-1}) x_o + g_o(y_{o+1},...,y_n) \end{cases}$$

여기에서 $r_1,...,r_o$는 영이 아닌 F_q에서의 임의의 원소이고, g_1,g_2,g_3은 임의의 3차 다항식이다. 함수 F의 구조를 숨기기 위해서 가역인 아핀변환 $S: F_q^n \rightarrow F_q^n$를 선택해서 $P = \mathrm{F} \circ S$를 공개키로 하고 함수 F와 아핀변환 S를 비밀키로 한다. 그러면 공개키는 3개의 3차 다변수 다항식과 $o-3$개의 2차 다변수 다항식을 컴포넌트로 가지게 된다.

서명생성: 서명 $\sigma \in F_q^n$을 생성하기 위해서 평문의 해쉬값 $d = (d_1,...,d_o) \in F_q^o$에 대해서 다음을 실행한다.

- 임의의 값들을 선택해서 다항식 $\hat{f}^{(1)},...,\hat{f}^{(o)}$와 $g_1,...,g_o$의 vinegar 변수 $y_{o+1},...,y_n$에 대입한다.
- 다음과 같이 $x_1,x_2,...,x_o$를 계산한다.

$$\begin{aligned} x_1 &= \frac{1}{r_1} \cdot (d_1 - g_1) - \frac{1}{r_2} \cdot (d_2 - g_2) \\ x_2 &= \frac{1}{r_2 \cdot x_1} \cdot (d_2 - g_2) \\ &... \\ x_i &= \frac{1}{r_i \cdot (x_{i-2} + x_{i-1})} \cdot (d_i - g_i) \quad (i = 3,...,o) \end{aligned}$$

만약 위의 식의 분모가 0이 되면 첫 번째 단계에서의 임의의 값을 다시 선택한다.

- 첫번째 단계에서 만들어진 마지막 $o-1$개의 식에 의해서 주어진 일차연립방정식을 풀어서 $y_2,...,y_o$를 y_1에 대한 일변수 일차식으로 표현한다. 만약 해가 없으면 첫 번째 단계를 다시 반복한다.
- 위의 단계에서 얻어진 $y_2,...,y_o$를 $\hat{f}^{(1)}$에 대입하여 y_1에 대한 일변수 다항식을 얻고 이것을 풀어서 y_1의 값을 구한다.
- $\sigma = S^{-1}(y_1,...,y_n)$을 계산하여 서명 σ를 얻는다.

서명검증: **주**어진 서명 σ와 평문의 해쉬값 d에 대해서 $P(\sigma)$와 d가 같은지 다른지를 확인하여 같으면 올바른 서명으로 확인한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

(나) CUOV에 대한 Hashimoto의 공격

CUOV의 중심 함수 F의 첫 번째와 두 번째 컴포넌트 함수 $\overline{f}^{(1)}$, $\overline{f}^{(2)}$는 $\overline{f}^{(1)} = r_1 \cdot (x_1 + x_1 x_2) + g_1(y_{o+1},...,y_n)$, $\overline{f}^{(2)} = r_2 \cdot x_1 x_2 + g_2(y_{o+1},...,y_n)$와 같이 정의되어 있음을 알 수 있다. 따라서 $\overline{f}^{(1)} - r_1 r_2^{-1} \cdot \overline{f}^{(2)}$를 계산하면 $x_1 x_2$부분이 소거되어서 모든 변수에 대한 이차다항식과 $y_{o+1},...,y_n$에 대한 삼차다항식 부분만이 남음을 알 수 있다. 이러한 성질을 이용하여 CUOV의 비밀키 S을 복원할 수 있고 공개키와 이것을 이용하여 나머지 비밀키 F도 복원할 수 있는 것이다. 이러한 Hashimoto의 공격은 다음과 같다.

• $x = (x_1,...,x_n)$에 대해서 공개키 $P(x) = (f_1(x),...,f_o(x))$라 하자. 그러면 임의의 상수 $c \in F_q^n$를 선택하여 $D_c f_i(x) = f_i(x+c) - f_i(x)$ $(i = 1,2)$를 계산한다. 그리고 $D_c f_i(x)$의 이차항들의 계수 행렬을 Q_i라 놓는다.

• $Q_1 + \beta Q_2$의 랭크가 v이하가 되는 상수 β를 찾는다. $\overline{f}^{(1)} - r_1 r_2^{-1} \cdot \overline{f}^{(2)}$의 삼차다항식부분이 변수 $y_{o+1},...,y_n$로 이루어져 있기 때문에 이러한 β는 존재한다. 또한 공개키의 만든 방식에 의해서 찾은 $Q_1 + \beta Q_2$는 다음과 같은 형태를 가지게 됨을 알 수 있다.

$$Q_1 + \beta Q_2 = S^t \begin{pmatrix} 0_o & 0 \\ 0 & *_v \end{pmatrix} S$$

• 다음을 만족하는 $v \times o$ 행렬 M을 찾는다.

$$\begin{pmatrix} I_o & M^t \\ 0 & I_v \end{pmatrix} (Q_1 + \beta Q_2) \begin{pmatrix} I_o & 0 \\ M I_v \end{pmatrix} = \begin{pmatrix} 0_o & 0 \\ 0 & *_v \end{pmatrix}$$

• 그러면 $\begin{pmatrix} I_o & 0 \\ M I_v \end{pmatrix}$가 S^{-1}와 같은 역할을 함을 알 수 있고 이것과 공개키 P를 이용하면 정당하지 않은 사용자도 올바른 서명을 위조할 수 있게 된다.

(다) CUOV의 변형 CSSv와 SVSv

위의 CUOV에 대한 Hashimoto의 공격이 가능했던 이유는 공개키의 일차결합으로 랭크 n을 가지는 다변수 이차함수와 랭크 v를 가지는 다변수 삼차함수의 합의 형태를 가지는 함수를 얻을 수 있었다는 점이다. 이러한 공격을 막기 위하여 Duong 등은 CUOV의 두 가지 변형인 Cubic Signature Scheme with Vinegar(CSSv)와 Simple Vector Signature Scheme with Vinegar(SVSv)를 제안하였다. 먼저 CSSv를 살펴보면 다음과 같다.

① CSSv

F_q를 원소의 개수가 q개인 유한체라 하고, 자연수 o,v에 대해서 $n = o + v$를 변수의 개

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

수, o를 식의 개수라고 하자. 그러면 CSSv는 $\mathrm{F}: F_q^n \to F_q^o$인 함수 F, $T: F_q^o \to F_q^o$와 $S: F_q^n \to F_q^n$인 가역인 아핀변환 S, T에 대해서 $P = T \circ \mathrm{F} \circ S$를 공개키로 함수 F와 아핀변환 S, T를 비밀키로 가지는 전자서명이다. 구체적으로 함수 F는 다음과 같이 만들어진다.

키생성: 두 함수 $\overline{F}$와 $\hat{F} \times id_v$에 대해서 $\mathrm{F} = \overline{F} \circ (\hat{F} \times id_v)$로 정의된다. 먼저 $\hat{F} = (\hat{f}^{(1)}, ..., \hat{f}^{(o)}) : F_q^n \to F_q^o$는 다음과 같이 정의된다.

$$\begin{cases} \hat{f}^{(1)} = \sum_{i=1}^{n}\sum_{j=i}^{n} a_{ij}^{(1)} y_i y_j + \sum_{i=1}^{n} b_i^{(1)} y_i + c^{(1)} \\ \hat{f}^{(2)} = \sum_{i=1}^{n} b_i^{(2)} y_i + c^{(2)} \\ \cdots \\ \hat{f}^{(o)} = \sum_{i=1}^{n} b_i^{(o)} y_i + c^{(o)} \end{cases}$$

여기서 계수는 모두 F_q에서 임의로 선택된 원소이다. 그리고

$$\begin{aligned} \hat{F} \times id_v : F_q^n &\to F_q^n \\ (y_1, ..., y_o, y_{o+1}, ..., y_n) &\mapsto (\hat{f}^{(1)}, ..., \hat{f}^{(o)}, y_{o+1}, ..., y_n) \end{aligned}$$

로 정의된다. 함수 $\overline{F}$는 다음과 같이 정의된다.

$$\begin{aligned} \overline{F} : F_q^o \times F_q^v &\to F_q^o \\ (x_1, ..., x_o, y_{o+1}, ..., y_n) &\mapsto (\overline{f}^{(1)}, ..., \overline{f}^{(o)}) \end{aligned}$$

$\overline{f}^{(i)}$는 다음과 같다.

$$\begin{cases} \overline{f}^{(1)} = x_1 + g_1(y_{o+1}, ..., y_n), \\ \overline{f}^{(2)} = x_1 x_2 + g_2(y_{o+1}, ..., y_n), \\ \quad ... \\ \overline{f}^{(o)} = x_{o-1} x_o + g_o(y_{o+1}, ..., y_n) \end{cases}$$

여기에서 g_2는 임의의 3차 다항식이고, $g_1, g_3, ..., g_o$는 임의의 2차 다항식이다. 함수 F의 구조를 숨기기 위해서 가역인 아핀변환 $S: F_q^n \to F_q^n$와 다음의 구조를 가지는 아핀변환 $T: F_q^o \to F_q^o$를 선택한다.

$$T = \begin{pmatrix} *_{1\times 1} & *_{1\times 1} & *_{1\times(o-2)} \\ *_{(o-1)\times 1} & 0_{(o-1)\times 1} & *_{(o-1)\times(o-2)} \end{pmatrix} \in F_q^{o\times o}$$

이것으로부터 $P = T \circ \mathrm{F} \circ S$를 공개키로 하고 함수 F와 아핀변환 S, T를 비밀키로 한다. 그러면 공개키는 1개의 3차 다변수 다항식과 $o-1$개의 2차 다변수 다항식을 컴포넌트로

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

가지게 된다.

서명생성: 서명 $\sigma \in F_q^n$을 생성하기 위해서 평문의 해쉬값 $d=(d_1,...,d_o) \in F_q^o$에 대해서 다음을 실행한다.

• $w = T^{-1}(d)$를 계산한다.
• 임의의 값들을 선택해서 다항식 $\hat{f}^{(1)},...,\hat{f}^{(o)}$와 $g_1,...,g_o$의 vinegar 변수 $y_{o+1},...,y_n$에 대입한다.
• 다음과 같이 $x_1,x_2,...,x_o$를 계산한다.

$$x_1 = w_1 - g_1$$
$$x_i = \frac{1}{x_{i-1}} \cdot (w_i - g_i) \ \ (i=2,...,o)$$

만약 위의 식의 분모가 0이 되면 첫 번째 단계에서의 임의의 값을 다시 선택한다.
• 두번째 단계에서 만들어진 마지막 $o-1$개의 식에 의해서 주어진 일차연립방정식을 풀어서 $y_2,...,y_o$를 y_1에 대한 일변수 일차식으로 표현한다. 만약 해가 없으면 첫 번째 단계를 다시 반복한다.
• 위의 단계에서 얻어진 $y_2,...,y_o$를 $\hat{f}^{(1)}$에 대입하여 y_1에 대한 일변수 다항식을 얻고 이것을 풀어서 y_1의 값을 구한다.
• $\sigma = S^{-1}(y_1,...,y_n)$을 계산하여 서명 σ를 얻는다.

서명검증: 주어진 서명 σ와 평문의 해쉬값 d에 대해서 $P(\sigma)$와 d가 같은지 다른지를 확인하여 같으면 올바른 서명으로 확인한다.

② SVSv

F_q를 원소의 개수가 q개인 유한체라 하고, 자연수 o,v,r에 대해서 $n=o+v+r$를 변수의 개수, o를 식의 개수라고 하자. 그러면 SVSv는 $\mathrm{F}: F_q^n \to F_q^o$인 함수 F, $T: F_q^o \to F_q^o$와 $S: F_q^n \to F_q^n$인 가역인 아핀변환 S,T에 대해서 $P = T \circ \mathrm{F} \circ S$를 공개키로 함수 F와 아핀변환 S,T를 비밀키로 가지는 전자서명이다. 구체적으로 함수 F는 다음과 같이 만들어진다.

키생성: 두 함수 $\overline{F}$와 $\hat{F} \times id_v$에 대해서 $\mathrm{F} = \overline{F} \circ (\hat{F} \times id_v)$로 정의된다. 먼저 $\hat{F} = (\hat{f}^{(1)},...,\hat{f}^{(o)}) : F_q^n \to F_q^o$는 다음과 같이 정의된다.

$$\hat{f}^{(i)} = \sum_{j=1}^{n} a_{ij} y_j + c_i \ \ (i=1,...,o)$$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

여기서 계수는 모두 F_q에서 임의로 선택된 원소이다. 그리고

$$\hat{F} \times id_v : F_q^n \rightarrow F_q^n$$
$$(y_1,...,y_o,y_{o+1},...,y_n) \mapsto (\hat{f}^{(1)},...,\hat{f}^{(o)},y_{o+1},...,y_n)$$

로 정의된다. 함수 $\overline{F}$는 다음과 같이 정의된다.

$$\overline{F} : F_q^o \times F_q^{v+r} \rightarrow F_q^o$$
$$(x_1,...,x_o,y_{o+1},...,y_n) \mapsto (\overline{f}^{(1)},...,\overline{f}^{(o)})$$

$\overline{f}^{(i)}$는 다음과 같다.

$$\begin{cases} \overline{f}^{(1)} = x_1^2 + g_1(y_{o+1},...,y_n), \\ \overline{f}^{(2)} = x_1x_2 + g_2(y_{o+1},...,y_{o+v}), \\ \quad ... \\ \overline{f}^{(o)} = x_{o-1}x_o + g_o(y_{o+1},...,y_{o+v}) \end{cases}$$

여기에서 $g_1,...,g_o$는 임의의 2차 다항식이다. 함수 F의 구조를 숨기기 위해서 가역인 아핀변환 $S: F_q^n \rightarrow F_q^n$와 아핀변환 $T: F_q^o \rightarrow F_q^o$를 선택한다. 이것으로부터 $P = T \circ F \circ S$를 공개키로 하고 함수 F와 아핀변환 S, T를 비밀키로 한다.

서명생성: 서명 $\sigma \in F_q^n$을 생성하기 위해서 평문의 해쉬값 $d = (d_1,...,d_o) \in F_q^o$에 대해서 다음을 실행한다.

- $w = T^{-1}(d)$를 계산한다.
- 임의의 값들을 선택해서 다항식 $\hat{f}^{(1)},...,\hat{f}^{(o)}$와 $g_1,...,g_o$의 vinegar 변수 $y_{o+1},...,y_n$에 대입한다.
- 다음과 같이 $x_1, x_2,...,x_o$를 계산한다.

$$x_1 = \sqrt{w_1 - g_1} = \begin{cases} (w_1 - g_1)^{1/2} & q \equiv 1 \bmod 2 \\ (w_1 - g_1)^{q/2} & q \equiv 0 \bmod 2 \end{cases}$$

만약 $x_1 = 0$이면 두 번째 단계로 돌아간다.

$$x_i = \frac{1}{x_{i-1}} \cdot (w_i - g_i) \quad (i = 2,...,o)$$

만약 위의 식의 분모가 0이 되면 두 번째 단계에서의 임의의 값을 다시 선택한다.

- 위의 단계에서 얻어진 $(x_1,...,x_o)$에 대해서 $\hat{f}^{(1)},...,\hat{f}^{(o)}$에 의해서 주어진 일차연립방정식을 풀어서 $(y_1,...,y_o)$을 구한다. 만약 해가 없으면 두 번째 단계로 돌아간다.
- $\sigma = S^{-1}(y_1,...,y_n)$을 계산하여 서명 σ를 얻는다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

서명검증: 주어진 서명 σ와 평문의 해쉬값 d에 대해서 $P(\sigma)$와 d가 같은지 다른지를 확인하여 같으면 올바른 서명으로 확인한다.

(5) HFEv- Signature Scheme

Petzoldt 등은 Asiacrypt 2015에서 새로운 서명 알고리즘 Gui를 발표하였다[PCYT15]. 논문 제목만 봐서는 HFE계열의 새로운 스킴이 발표된 것으로 보였으나 HFEv- 서명 스킴은 이미 오래 전부터 존재하여 널리 연구되던 스킴이었다. HFE(Hidden Field Equation) 서명 알고리즘은 1996년 Patarin에 의해 제안되었으나[P96], 이런 기본적인 형태는 여러 가지 공격에 안전하지 않은 것으로 드러났다[FJ03, KS99]. 그럼에도 HFE을 약간 변형하여 알려진 공격에 안전하도록 하는 스킴들이 발표되었다. HFEv- 스킴은 HFE에 vinegar 변형, -변형을 준 것으로 vinegar 변형은 기존의 HFE의 중앙 비밀함수에 vinegar 변수들을 추가로 섞어줘서 안전성을 높이는 방법이고, -변형은 공개키 함수의 좌표함수들 중 일부를 공개하지 않아서 현존하는 여러 공격들을 막는 방법이다. 이미 QUARTZ라는 HFEv- 기반의 서명 알고리즘이 발표되었으나[PCG01], 효율성이 너무 떨어져 실제로 사용되기에는 어려움이 있었다. 하지만, Ding과 Yang의 HFEv- 다항식에 대한 분석[DY13]을 기반으로 QUARTZ보다 효율적으로 파라미터를 조절하더라도 안전성이 충분하다는 사실과 몇 몇 실험 결과에 기반하여 다른 파라미터 적용한 HFEv-가 Gui 전자서명이다. [PCYT15]

(가) HFEv-

HFE에 대해 다루기 전에 먼저 canonical 동형사상부터 살펴보자. 이는 $\Phi: F_q^n \mapsto F_{q^n}$에서 정의되서 $\Phi(x_1,\cdots,x_n)=\sum_{i=1}^{n} x_i X^{i-1}$로 정의된다. 일반적인 HFE는 아래와 같은 유형의 다항식을 말한다.

$$f(X)=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\alpha_{ij}X^{q^i+q^j}+\sum_{i=0}^{n-1}\beta_{ij}X^{q^i}+\gamma$$

위의 식에서 $\alpha_{ij},\beta_i,\gamma\in F_{q^n}$이다. 하지만 HFEv-에서는 β_i,γ가 단지 상수가 아닌 vinegar 변수들에 관한 식으로 변한다. 따라서 아래와 같은 유형이 된다.

$$f(X)=\sum_{0<i\le j}^{q^i+q^j\le D}\alpha_{ij}X^{q^i+q^j}+\sum_{i=0}^{q^i\le D}\beta_i(v_1,\cdots,v_v)X^{q^i}+\gamma(v_1,\cdots,v_v)$$

여기서 $\alpha\in F_{q^n}$, $\beta_i: F_q^v \mapsto F_{q^n}$는 1차식, $\gamma: F_q^v \mapsto F_{q^n}$은 2차식이다. 비밀키 $F=\Phi^{-1}\circ f\circ\Phi$가 독특한 형태를 가지고 있으므로, 이 구조를 숨기기 위하여 F의 양쪽에 두 개의 affine 변환

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$S: F_q^n \mapsto F_q^{n-a}$, $T: F_q^{n+v} \mapsto F_q^{n+v}$를 배치한다. 공개키는 다른 스킴들과 마찬가지로 $P = S \circ F \circ T: F_q^{n+v} \mapsto F_q^{n-a}$이며 비밀키는 S, F, T이다.

서명생성: 메시지 $h \in F_q^{n-a}$에 대해 서명을 생성하고자 할 경우 먼저 $x = S^{-1}(h) \in F_q^n$를 계산한다. 그런 다음 $X = \Phi(x)$를 계산한 후 vinegar 변수인 $v_1, v_2, \cdots, v_v \in F_q$를 임의로 선택하여 F의 β_i, γ에 대입한다. 이제 1변수 다항식이 되었으므로, Berlekamp의 알고리즘을 이용하여 $F(Y) = X$가 되는 $Y \in F_{q^n}$을 찾아낸다. $y' = \Phi^{-1}(Y) \in F_q^n$를 계산한 후 $y = (y' \| v_1 \| \cdots \| v_v)$라고 하자. 마지막으로 $z = T^{-1}(y) \in F_q^{n+v}$를 계산하면 z가 메시지 h의 서명값이 된다.

서명검증: 서명값 $z \in F_q^{n+v}$가 메시지 h의 유효한 서명인지를 살펴보기 위해서는 다른 다변수 다항식 기반 서명 알고리즘과 마찬가지로 $P(z) = h' \in F_q^{n-a}$를 계산한 다음, $h = h'$이면 서명은 유효한 서명이고, 그렇지 않으면 이는 유효하지 않은 서명이다.

(나) QUARTZ

QUARTZ는 HFEv-의 아이디어를 이용한 서명 알고리즘이다[PCG01]. 이 서명 알고리즘의 독특한 점은 서명 값이 128 비트로 아주 작다는 점이다. QUARTZ에서는 $(q, n, D, a, v) = (2, 103, 129, 3, 4)$의 파라미터를 갖는 HFEv-를 적용하였다. 공개키의 길이는 71KB이고 비밀키의 길이는 3KB이다.

QUARTZ의 입력값의 길이는 겨우 $n - a = 100$비트밖에 되지 않는다. 따라서 공격자가 같은 입력값을 갖는 두 개의 메시지를 찾는 생일공격을 시도할 수 있다. 이런 종류의 공격을 예방하기 위해 QUARTZ의 서명 생성과정에 특별한 과정을 집어넣었다. 한 개의 메시지에 4개의 HFEv- 서명 값을 계산한 뒤 그것을 합쳐 128 비트의 서명값을 생성해내는 것이다. 물론 서명 확인과정에서도 공개키를 4번 사용하여야 한다.

이러한 HFEv- 전자서명 알고리즘에서 가장 많이 시간을 요구하는 과정은 역시 F의 역원을 계산하는 과정이다. 이 과정은 Berlekamp의 알고리즘을 통해서 계산되며 그 복잡도는 $O(D^3 + nD^2)$이다. 즉, HFEv- 서명 알고리즘의 복잡도는 결국 F의 차수인 D에 의해 결정된다. QUARTZ 알고리즘의 경우 D가 상대적으로 큰데다, F의 역상을 구할 때 해가 한개 뿐이다. 그리고 vinegar 변수를 고른 뒤에 $\frac{1}{e}$의 확률로 F가 random 함수처럼 보일 때가 있다. 이 때문에 QUARTZ에서는 이 과정에서 Berlekmap의 알고리즘을 약 $4e$번이나 돌려야 한다. 따라서 QUARTZ 서명 알고리즘은 상대적으로 느리며, 서명을 생성하는데 약 11초 정도가 걸린다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

(다) Gui

QUARTZ는 실제로 사용하기에 다소 무리가 있다. Petzoldt등은 HFEv-에 가장 주요한 두 가지 공격인 direct attack과 MinRank attack에 대한 안전성을 분석하여 D를 낮추는 대신 다른 파라미터를 조정하여 안전성은 유지하면서도 좀더 빠른 알고리즘을 설계하고자 했다[PCYT15].

우선 HFE에 대한 MinRank attack[KS99]에 대해 간단히 언급하겠다. 편의상 P,F가 homogeneous인 경우에 한정하도록 한다. 공격의 핵심 아이디어는 S,T,P를 F_{q^n}에서의 함수 $S^\star, T^\star, P^\star$로 확장한 것이다. 이 경우 S,T는 일차함수이므로, $S^\star$, $T^\star$는

$$S^\star(X)=\sum_{i=1}^{n-1} s_i \cdot X^{q^i},\ \ T^\star(X)=\sum_{i=1}^{n-1} t_i \cdot X^{q^i} (s_i,t_i \in F_q)$$

의 형태를 갖는다. 공개키 $P^\star$는

$$P^\star(X)=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} p_{ij}^\star X^{q^i+q^j}=\underline{X}\cdot P^\star \cdot \underline{X^{\mathrm{T}}}$$

의 형태를 갖는데 여기서 $P^\star=(p_{ij}^\star)_{1\le i,j\le n}$이고 $\underline{X}=(X^{q^0},X^{q^1},\cdots,X^{q^{n-1}})$이다. 그런데 $P^\star(X)=(S^\star \circ F \circ T^\star)(X)$를 만족하므로 $(S^{\star -1}\circ P^\star)(X)=(F\circ T^\star)(X)$가 된다. 그리고 $G^{\star k}=(g_{ij}^{\star k})_{1\le i,j\le n}=(p_{i-k \bmod n, j-k \bmod n}^\star)_{1\le i,j\le n}^{q^k}$, $W=(w_{ij})_{1\le i,j\le n}=(s_{j-i \bmod n}^{q^i})_{1\le i,j\le n}$에 대해 $\tilde{P}=\sum_{k=0}^{n-1} s_k G^{\star k}=W\cdot F\cdot W^{\mathrm{T}}$가 된다. F의 특별한 구조로 인해 F는 왼쪽 위의 $r\times r$ 부분행렬($r=\lfloor \log_q D-1 \rfloor+1$)에만 존재한다. 따라서 행렬 $W\cdot F\cdot W^{\mathrm{T}}$의 rank는 r보다 작거나 같다. 이는 MinRank 문제를 이용하여 s_k를 찾을 수 있다는 뜻이다. HFEv-의 경우에는 $\tilde{P}$의 rank는 $r+a+v$보다 같거나 작다. 이 경우 MinRank 공격에 대한 복잡도는 $O(q^{n(r+v+a-1)}\cdot(n-a)^3)$이다. 따라서 우리가 $r+a+v$의 값을 늘릴 경우 이 공격에 대해 안전하게 된다.

HFEv-에 대한 direct attack의 복잡도는 QUARTZ가 제안될 때만 해도 잘 알려져 있지 않았다. 따라서 QUARTZ의 저자들은 이 공격방법에 대한 이론적인 복잡도를 제시하지 못하였으며, 여러 실험 결과 HFEv-에 기반한 서명 알고리즘들은 random한 다변수 시스템에 비해 쉽게 풀린다는 사실을 알 수 있었다. Ding과 Yang이 [DY13]에서 HFEv-에 대한 degree of regularity의 한 상계가 아래와 같음을 증명하였다.

$$d_{reg}\le \begin{cases} \dfrac{(q-1)\cdot(r+a+v-1)}{2}+2 & q:even \text{ and } r+a:odd \\ \dfrac{(q-1)(r+a+v)}{2}+2 & otherwise \end{cases}$$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

여기서 $r = \lfloor \log_q(D-1) \rfloor + 1$이다.

이제 앞에서 고려해본 공격방법들에 대한 공격복잡도와 다음 사항들을 고려하여 실험한 후 파라미터를 정할 수 있다.

① 위의 degree of regularity에 대한 정보를 이용하여 HFE의 차수 D를 낮추더라도 전체 스킴이 안전하도록 할 수 있는가?

▷ 실험결과 $D=5,9,17$ 정도로 작게 선택할 수 있었다. 단, d_{reg}가 $r+a+v$에 의존하기 때문에 r이 작아지면 그만큼 $a+v$를 키우면 된다.

② a와 v의 비율이 스킴의 안전성에 중요한 영향을 주는가?

▷ 실험결과 v가 너무 작아서도 안되지만, 너무 커지더라도 안전성을 키우지는 않는다. 따라서 키 길이를 줄이기 위해 $v-a \leq 1$이 될 정도로 a,v를 비슷하게 잡을 것이다.

③ 위의 d_{reg} 정보는 tight한가?

▷ 식의 개수를 너무 키울 수 없어 정확한 실험을 할 수는 없었다. 하지만 실험결과 d_{reg}가 최소 7을 달성하는 파라미터들을 뽑았다.

④ Gröbner basis 알고리즘을 구현할 때 몇 개의 변수를 추측하는 방법이 스킴의 안전성에 어떤 영향을 주는가?

▷ 실험 결과 그런 아이디어를 쓸 경우 식의 개수가 더 많아야 d_{reg}의 최소값이 7이 되도록 할 수 있었다. 따라서 n을 충분히 크게 선택하여 이런 공격에 안전하도록 하겠다.

위의 사항들을 고려하여 아래와 같은 파라미터를 설정하였다.

- Gui-96 : $(n,D,a,v) = (96,5,6,6)$
- Gui-95 : $(n,D,a,v) = (95,9,5,5)$
- Gui-94 : $(n,D,a,v) = (94,17,4,4)$

위의 3개의 파라미터는 80 비트 정도의 안전성을 갖는다. $(n,D,a,v) = (127,9,4,6)$인 Gui-127은 120 비트의 안전성을 갖는다.

전체적인 서명 생성 및 서명 확인 과정은 앞에서 소개한 HFEv-의 서명 생성 및 서명 확인 과정과 유사하다. 다만 QUARTZ에서도 그랬듯이 입력값의 길이가 겨우 $n-a=90$ 비트이기 때문에 생일 공격을 받을 수 있다. 따라서 서로 다른 입력값에 대해 이러한 서명 과정을 여러번 실행할 것인데 이 반복하는 횟수를 k라고 하면, Gui-96과 Gui-95에는 $k=3$을, Gui-94와 Gui-127에는 $k=4$를 적용한다.

서명생성: 전체 서명을 생성하기 위해 먼저 $n-a$ 벡터 $S_0 = 0$이라고 한 후 메시지의 해쉬값 h을 계산한다. D_1을 h의 처음 $(n-a)$비트라고 하자. D_1의 역원을 HFEv-의 역연산을 적용하여 구한 뒤 처음 $(n-a)$비트를 S_1, 다음 $a+v$비트를 X_1이라고 한다. 이제 h에 한 번 더 해쉬함수를 적용한 값의 처음 $(n-a)$비트를 D_2라고 한 뒤 $D_2 \oplus S_1$의 역원을 위와 마찬가지로 구하여 처음 $(n-a)$비트를 S_2, 다음 $a+v$비트를 X_2라고 한다. 이 과정을

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

S_k, X_k를 구할 때까지 계속한다. 마지막 서명 값은 $\sigma = (S_k \| X_k \| \cdots \| X_1)$이다.

서명검증: 서명을 확인하기 위해서는 먼저 서명값 σ로부터 $S_k, X_k, \cdots, X_1$값을 얻어낸 후, $D_1, \cdots, D_k$값을 해쉬함수를 통해 구해낸다. $S_i = P(S_{i+1} \| X_{i+1}) \oplus D_{i+1}$을 귀납적으로 구한 뒤 $S_0 = 0$이면 서명을 받아들이고, 아니라면 거부한다.

이러한 Gui에 대해 F의 역연산을 하는 시간은 QUARTZ에 비해 훨씬 짧은 시간이 걸리며, 따라서 서명하는데 걸리는 시간 역시 굉장히 짧다. 반면, 서명확인 시간은 크게 차이나지는 않는다. 또한 이러한 Gui의 퍼포먼스는 현재 사회에서 널리 사용되고 있는 RSA 서명 알고리즘이나, ECDSA와 비교해도 크게 뒤지지 않는다. 자세한 데이터는 [PCYT15]에서 참고하면 된다.

다. 다변수 이차식 기반 암호화 알고리즘

(1) Simple Matrix Encryption Scheme

Simple Matrix Encryption Scheme은 Post-Quantum Cryptography 2013(PQCrypto 2013) 학회에서 발표된 다변수 다항식 기반 암호화 알고리즘이다[TDTD13]. 서명 알고리즘의 경우 UOV, Rainbow 등 아직 깨지지 않고 남아있는 알고리즘들이 몇 개 있지만, 암호화 알고리즘의 경우 거의 모든 알고리즘이 깨져 남아있는 알고리즘이 없다. 이러한 상황에서 소개된 이 암호화 알고리즘은 중앙의 비밀키를 정의하기 위해 정사각행렬 3개를 이용하기 때문에 ABC 스킴이라고도 불린다. 이 암호화 알고리즘은 복호화 실패(decryption failure) 확률이 높은데다, 비교적 큰 유한체를 사용하여 느리다는 단점이 있지만, 빠른 구현을 통해 현존하는 다른 공개키 암호가 갖고 있는 성능과 비슷한 성능을 갖도록 구현되었다[PTCW16]. 이 암호화 알고리즘을 깼다는 결과가 최근에 제시되었으나 [Gu16], 공격복잡도가 다소 높은 편이므로, n을 키우면 이 공격에도 충분한 안전성을 가질 것으로 보인다.

(가) ABC Scheme

이 암호화 알고리즘 역시 두 개의 affine 변환 $S: F_q^m \mapsto F_q^m$, $T: F_q^n \mapsto F_q^n$과 2차 변환 $F: F_q^n \mapsto F_q^m$를 비밀키로 하고 $P = S \circ F \circ T$를 공개키로 한다는 사실은 서명 알고리즘과 비슷한 사항이다. 하지만, 메시지에 대응하는 암호문이 반드시 존재해야 하기 때문에 암호문들의 집합인 F_q^m를 메시지의 집합인 F_q^n보다 훨씬 크게 하여 $m = 2n > n$을 만족하도록 하였다. 보통 $m \leq n$을 만족하는 서명 알고리즘들과는 다르다. 또한 n개의 항을 갖는 정사각행렬이 정의되어야 하기 때문에 n은 반드시 제곱수여야 한다. 즉, $n = s^2$를 만족하

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

는 양의 정수 s가 존재한다. 이제 중앙의 2차 비밀 함수를 정의하기 위해 3개의 행렬을 살펴본다.

$$A=\begin{pmatrix} x_1 & x_2 & \cdots x_s \\ x_{s+1} & x_{s+2} & \cdots x_{2s} \\ \vdots & \vdots & \ddots \vdots \\ x_{s(s-1)+1} & x_{s(s-1)+2} & \cdots x_n \end{pmatrix}, B=\begin{pmatrix} b_1 & b_2 & \cdots b_s \\ b_{s+1} & b_{s+2} & \cdots b_{2s} \\ \vdots & \vdots & \ddots \vdots \\ b_{s(s-1)+1} & b_{s(s-1)+2} & \cdots b_n \end{pmatrix}, C=\begin{pmatrix} c_1 & c_2 & \cdots c_s \\ c_{s+1} & c_{s+2} & \cdots c_{2s} \\ \vdots & \vdots & \ddots \vdots \\ c_{s(s-1)+1} & c_{s(s-1)+2} & \cdots c_n \end{pmatrix}$$

A의 각 성분은 1개의 변수로 이루어져있다. B와 C의 각 성분은 상수로 되어있는 것처럼 보이지만 실제로는 각 $b_i, c_j (1 \le i,j \le n)$들은 $x_1, \cdots, x_n$의 1차 결합으로 되어있다. 이제 $E_1 = AB$, $E_2 = AC$라고 정의하자. 각 b_i, c_j가 1차 다항식이므로 E_1, E_2의 각 항은 다변수 2차식으로 되어있다. 이제 E_1, E_2의 각 성분을 index의 사전식 순서대로 좌표로 나열하였다고 생각하면 이는 F_q^n에서 $F_q^{2n} = F_q^m$으로 가는 하나의 다변수 2차식 함수가 된다. 이 함수를 F라고 정의한다. 이것이 ABC 암호화 알고리즘의 키 생성 방법이다.

■ 키 생성 방법

① 임의의 선형 변환 $S: F_q^m \mapsto F_q^m$, $T: F_q^n \mapsto F_q^n$를 선택하고, 두 s차 정사각행렬 B, C의 각 다항식의 계수를 선택한다.

② $E_1 = AB, E_2 = AC$를 계산하여 F를 계산한 후, $P = S \circ F \circ T$를 계산한다.

③ P가 공개키, S, F, T는 비밀키이다.

■ 암호화 방법

암호화하고자 하는 메시지 $M=(M_1, \cdots, M_n)$에 대해 $(C_1, C_2, \cdots, C_m) = P(M_1, M_2, \cdots, M_n)$를 계산하면 $C=(C_1, C_2, \cdots, C_m) \in F_q^m$이 메시지 M의 암호문이다.

복호화 방법을 설명하기에 앞서 어떻게 복호화가 이루어지는지 살펴보겠다. S, T는 선형 변환이므로 전단사함수이다. 따라서 자연스럽게 역변환을 취해주면 된다. 하지만 F의 역변환은 그리 간단하게 여겨지지는 않는다. 암호문 $C=(C_1, C_2, \cdots, C_m) \in F_q^m$에 대해 $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}) = S^{-1}(C_1, C_2, \cdots, C_m) \in F_q^m$를 계산한 후 $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}) = F(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$를 만족하는 $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$를 찾기 위해 어떠한 방법을 사용하는지를 살펴본다. $\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}$를 순서대로 한 행에 s개씩 나열하여 s차 정사각행렬 2개를 만든 후 $\overline{E_1}, \overline{E_2}$라고 하고, A, B, C의 각 x_i 자리에 $\overline{x_i} (1 \le i \le n)$를 대입한 함수를 $\overline{A}, \overline{B}, \overline{C}$라 하면 $\overline{E_1} = \overline{A}\,\overline{B}$, $\overline{E_2} = \overline{A}\,\overline{C}$를 만족하므로, 이를 이용하여 아래의 3가지 방법을 고려해본다.

(i) $\overline{E_1}$이 가역이라면, $\overline{E_1}^{-1} = \overline{B}^{-1}\overline{A}^{-1}$이므로 $\overline{B}\overline{E_1}^{-1}\overline{E_2} = \overline{B}\overline{B}^{-1}\overline{A}^{-1}\overline{A}\,\overline{C} = \overline{C}$이다. 즉,

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$는 $B\overline{E_1}^{-1}\overline{E_2} - C = 0$의 한 해이므로, $B\overline{E_1}^{-1}\overline{E_2} - C = 0$의 해를 구한다.

(ii) $\overline{E_1}$이 가역이 아니지만 $\overline{E_2}$가 가역이라면 위의 (i)과 비슷한 분석을 통해 $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$가 $C\overline{E_2}^{-1}\overline{E_1} - B = 0$의 한 해이므로, $C\overline{E_2}^{-1}\overline{E_1} - B = 0$의 해를 구한다.

(iii) $\overline{E_1}, \overline{E_2}$ 모두 가역이 아니지만, $\overline{A}$가 가역일 경우 $\overline{E_1} = \overline{A}\overline{B}, \overline{E_2} = \overline{A}\overline{C}$의 양변에 $\overline{A}$의 역행렬을 취한 $\overline{A}^{-1}\overline{E_1} = \overline{B}$, $\overline{A}^{-1}\overline{E_2} = \overline{C}$로부터 $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$가 $\overline{A}^{-1}\overline{E_1} - \overline{B} = 0$, $\overline{A}^{-1}\overline{E_2} - \overline{C} = 0$의 공통 해가 된다는 사실을 알 수 있다. 따라서 $\overline{A}^{-1}\overline{E_1} - B = 0$, $\overline{A}^{-1}\overline{E_2} - C = 0$로부터 만들어진 연립방정식 시스템을 살펴보면, $\overline{A}$의 각 성분을 모르고 있기 때문에 이들까지 변수로 둔다면 우리는 여기서 $2n$개의 식과 변수를 가진 1차 연립방정식 시스템을 얻는다.

(iv) 위의 3가지에 해당하지 않을 경우 복호화 실패로 간주한다.

이제 복호화 과정을 정리해보자.

■ 복호화 방법

① $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}) = S^{-1}(C_1, C_2, \cdots, C_m)$를 계산한 후, $\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}$를 순서대로 한 행에 s개씩 나열하여 s차 정사각행렬 2개를 만든 후 $\overline{E_1}, \overline{E_2}$라고 한다.

② 이제 $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}) = F(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$을 만족하는 $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$를 찾고자 한다.

(i) $\overline{E_1}$이 가역이라면, $B\overline{E_1}^{-1}\overline{E_2} - C = 0$의 해를 구한다.

(ii) $\overline{E_1}$이 가역이 아니지만 $\overline{E_2}$가 가역이라면 $C\overline{E_2}^{-1}\overline{E_1} - B = 0$의 해를 구한다.

(iii) $\overline{E_1}, \overline{E_2}$ 모두 가역이 아니지만, $\overline{A}$가 가역일 경우 $\overline{A}^{-1}\overline{E_1} - B = 0$, $\overline{A}^{-1}\overline{E_2} - C = 0$에서 추가로 $\overline{A}$의 성분까지 변수로 하는 $2n$개의 식과 변수를 가진 1차방정식 시스템의 해를 구한다.

(iv) 위의 3가지에 해당하지 않을 경우 복호화 실패로 한다.

이 스킴의 가장 중요한 두 가지 문제점은 복호화 실패의 확률이 크다는 점과 복호화 때 시간이 오래 걸린다는 것이다. 복호화 실패는 $\overline{A}$가 가역이 아닐 경우 나타나게 되는데 이렇게 될 확률은 $\frac{1}{q}$로 상대적으로 높은 편이다. 또한 복호화 시간이 오래 걸린다는 점은 위의 복호화 과정에서 살펴볼 수 있다. F의 역연산을 하는 부분 중에 (i) 부분에서 $B\overline{E_1}^{-1}\overline{E_2} - C = 0$의 해를 구하게 된다. 이는 해와 식의 개수가 같은 $Ax = 0$꼴의 1차 연립방정식을 푸는 꼴이다. 만약 이 연립방정식의 계수행렬이 가역이라면 이 방정식은 $x = 0$이라는 유일한 해를 갖게 된다. T도 선형 변환이기 때문에 결국 이는 메시지가 0이라는 것을 뜻한다. 항상 이렇게 될 수는 없으므로 결국 위의 연립방정식은 1개가 아닌 많은 해를 갖게 된다. 이 경우 모든 해에 대해 T의 역연산을 하여 메시지를 구한 뒤, 다시 암호

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

화를 하여 원래의 암호문이 나오는지를 확인하여야 한다. 이러한 과정을 모두 거치려면 복호화하는데 드는 시간이 늘어날 수밖에 없다. 이는 (ii), (iii)를 통해 해를 구하게 되는 경우 또한 마찬가지다.

(나) ABC 암호화 스킴의 향상

이러한 두 가지 단점을 보완하기 위하여 행렬 A,B,C를 정사각행렬이 아닌 행렬의 모양으로 설정하도록 ABC 암호화 스킴을 변형하는 방법이 제안되었다[TXPD15]. 변형된 방법에서는 A,B,C의 크기와 복호화 방법이 기존 ABC 암호화 스킴과 다르다. 먼저, A,B,C의 모양은 아래와 같다.

$$A=\begin{pmatrix} a_{1,1}\, a_{1,2} \cdots a_{1,r} \\ a_{2,1}\, a_{2,2} \cdots a_{2,r} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ a_{s,1}\, a_{s,2} \cdots a_{s,r} \end{pmatrix}, B=\begin{pmatrix} b_{1,1}\, b_{1,2} \cdots b_{1,u} \\ b_{2,1}\, b_{2,2} \cdots b_{2,u} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ b_{r,1}\, b_{r,2} \cdots b_{r,u} \end{pmatrix}, C=\begin{pmatrix} c_{1,1}\, c_{1,2} \cdots c_{1,v} \\ c_{2,1}\, c_{2,2} \cdots c_{2,v} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ c_{r,1}\, c_{r,2} \cdots c_{r,v} \end{pmatrix}$$

즉, A는 $s\times r$행렬, B는 $r\times u$행렬, C는 $r\times v$행렬이다. 따라서 $m=s(u+v)$가 된다. 물론 $r\le s$이므로 A의 항의 개수는 전체 변수의 개수보다 작다. 따라서 A의 각 항에는 $x_1,\cdots,x_n$을 임의로 배치해준다. $b_{i,j}$와 $c_{i,j}$에 $x_1,\cdots,x_n$의 일차 결합이 들어가게 된다는 것은 기존 ABC 암호화 스킴과 같다.

이제 이 새로운 스킴에서 어떻게 F의 역원을 찾는지 알아보도록 하겠다. 앞에서와 마찬가지로 암호문 $C=(C_1, C_2, \cdots, C_m)\in F_q^m$에 $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m})=S^{-1}(C_1, C_2, \cdots, C_m)\in F_q^m$를 계산한 후 $(\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m})=F(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$를 만족하는 $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$를 찾기 위해 어떠한 방법을 사용하는지를 살펴본다.

$\overline{y_1}, \overline{y_2}, \cdots, \overline{y_m}$를 순서대로 한 행에 s개씩 나열하여 $s\times u$행렬 한 개와 $s\times v$행렬 한 개를 만든 후 각각 $\overline{E_1}, \overline{E_2}$라고 하고, A,B,C의 각 x_i 자리에 $\overline{x_i}(1\le i\le n)$를 대입한 함수를 $\overline{A},\overline{B},\overline{C}$라 한다. 만약 $\overline{A}$의 $rank$가 r이라면, $W\overline{A}=I_r$을 만족하는 $r\times s$행렬 W가 존재한다. $\overline{E_1}=\overline{A}\overline{B}, \overline{E_2}=\overline{A}\,\overline{C}$의 양변의 왼쪽에 W를 곱하면, $W\overline{E_1}=W\overline{A}\overline{B}=\overline{B}$, $W\overline{E_2}=W\overline{A}\,\overline{C}=\overline{C}$를 얻는다. 즉, $(\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n})$는 $W\overline{E_1}=B$, $W\overline{E_2}=C$의 해다. W가 알려져있지 않으므로 W의 각 항들을 변수로 추가하면 $r(u+v)$개의 식과 $sr+n$개의 변수를 갖는 1차 연립방정식 시스템을 얻는다. W에 있는 sr개의 변수를 제거하면, $r(u+v-s)$개의 식과 $x_1,x_2,\cdots,x_n$의 변수로 이루어져 있는 시스템을 얻는다. 저자들에 의하면 이 시스템의 해공간의 차원은 아주 작으며, Gauss 소거법을 통해 많은 변수들을 제거할 수 있다고 한다. 만약 Z개의 변수들을 제거할 수 있도록 하면, 이 Z개의 변수들을 다른 변수들의 일차결합으로 표현한 다음 다른 식들에 대입한다. 그렇게 하면 m개의 식과 나머지 $n-Z$개의 변수로 이루어진 새로운 1차 연립방정식 시스템을 얻는다. 만약

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$(n-r(u+v-s))(n-r(u+v-s)+1) \leq 2m$$

을 만족하면, 재선형화 알고리즘(relinearization algorithm)을 이용하여 효과적으로 해를 찾을 수 있다. 하지만 만약 $Rank(\overline{A}) < r$이면, 복호화는 여전히 실패하게 된다. 하지만 이러한 복호화 실패 확률은 약 $\frac{1}{q^{s-r+1}}$이므로 파라미터 조절을 통해 이 확률을 무시할만할 정도로 낮출 수 있다. 실제로 복호화 실패 확률을 $\frac{1}{2^{32}}$ 이하로 낮추고, $u=v$를 만족하도록 하는 파라미터로 저자들은 표 9와 같은 파라미터를 제안하였다.

q	(r,s,u,v,m,n)
2^{32}	$(r, r+1, r, r, 2r(r+1), r(r+1))$
2^{16}	$(r, r+2, r+2, r+2, 2(r+2)^2, (r+2)^2)$
2^8	$(r, r+3, r+4, r+4, 2(r+3)(r+4), r(r+8))$

표 9. Simple Matrix Encryption Scheme의 파라미터 추천

저자들은 $r=8$또는 $r=9$를 사용하였다. 향상된 스킴은 기존의 ABC 암호화 스킴에 비해 복호화 실패 확률을 줄이고 키 길이를 줄였으며, 복호화 과정을 빠르게 하였다.

(다) Tensor를 이용하여 복호화 실패를 없애는 방법과 그 방법의 취약성

Petzoldt 등은 [PDW16]에서 Tensor를 이용하여 ABC 암호화 스킴의 복호화 실패를 없애는 방법에 대해 서술하였다. 이 방법의 핵심 아이디어는 T를 독특한 방법으로 설계하여 복호화 실패 여부를 암호화하기 전에 미리 알도록 하는 것이다. 암호화하는 사람은 해당 메시지를 암호화한 암호문을 복호화할 때 복호화 실패가 나타날지의 여부를 미리 알 수 있어, 암호문을 적당히 수정하여 복호화 실패가 나타나지 않도록 할 수 있다. 핵심 아이디어는 Tensor Product를 이용하는 것이다. 두 개의 가역인 $s \times s$행렬 T_1, T_2에 대해 두 행렬의 Tensot Product를 $T = T_1 \otimes T_2$로 설정하면 T 역시 가역이 된다. 이러한 T가 어떠한 성질을 갖는지 아래의 보조정리에서 소개한다.

보조정리 2. 임의의 $M=(M_1, \cdots, M_n) \in F_q^n (n=s^2)$에 대해 M의 각 성분을 차례로 s개씩 나열하여 s차 정사각행렬로 나타낸 것을 $A(M)$이라고 하자. 이 때 $A(M)$이 가역이면 $A(T(M))$ 역시 가역이다.

증명. $A(T(M)) = T_1 A(M) T_2^t$를 만족한다는 사실이 알려져 있다. 그런데 $T_1, A(M), T_2^t$ 모두 가역이므로 가역행렬의 곱인 $A(T(M))$ 역시 가역이다. ■

위의 보조정리에서 나온 $A(T(M))$값은 (1)의 ABC 암호화 스킴에서의 $\overline{A}$를 말한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

ABC 암호화 스킴에서는 $\overline{A}$가 가역이면 복호화 실패가 나타나지 않는다. 따라서 위 보조정리를 이용하여 암호화하고자 하는 사람은 메시지에 대해 $A(M)$이 가역임을 미리 점검한 후에 가역이 아닐 경우 의미가 변하지 않도록 약간 수정하여 $A(M)$이 가역이 되는 M을 골라서 암호화를 하게 되면 복호화 실패가 나타나지 않게 된다. 이런 사항을 적용하여 수정된 새로운 스킴을 Tensor Simple Matrix Encryption Scheme(TSMES)이라고 부르기로 한다. TSMES에 대한 direct attack의 경우 작은 파라미터들에 대한 실험을 통해 비교한 결과 같은 식과 변수를 갖는 ABC 암호화 스킴에 비해 떨어지지 않는다. 또한 이렇게 설계할 수 있는 T의 경우의 수는 굉장히 많으므로 ABC 암호화 스킴에 비해 그 안전성이 떨어지지 않는다고 저자들은 주장하였다.

하지만, 이 TSMES 알고리즘이 발표되자마자, TSMES가 가진 안전성이 SMES에 비해 떨어지지 않는다는 생각이 틀리다는 결과가 발표되었다[Has16]. 저자들은 표기의 편의성을 위해 두 가지 맵을 먼저 소개하였다.

$$\varphi(x_1, \cdots, x_n) = (x_{i+(s-1)j})_{1 \le i,j \le s},\ \psi\big((y_{i+(s-1)j})_{1 \le i,j \le s}, (y_{n+i+(s-1)j})_{1 \le i,j \le s}\big) = (y_1, \cdots, y_m)^t$$

φ는 $x_1, \cdots, x_n$을 한 행에 s개씩 차례로 s차 정사각행렬에 배치하는 함수이다. 임의의 $x \in F_q^n$를 A에 배치하는 것을 표현하려면 단지 $\varphi(x)$로 표현하면 충분하다. 한편, ψ는 E_1, E_2로 계산된 두 개의 s차 정사각행렬로부터 한 개의 F_q^m의 벡터를 얻어내는 함수이다. S변환을 시행하기 전에 E_1, E_2로 계산되어있는 두 개의 s차 정사각행렬을 F_q^m의 벡터로 만들어주려면 $\psi(E_1, E_2) \in F_q^m$으로 표현하면 충분하다. 이 경우 ABC 암호화 스킴의 공개키는 다음과 같이 표현된다.

$$P(M) = S\psi(\varphi(TM)\varphi(BTM), \varphi(TM)\varphi(CTM))$$

한편 비밀키가 S, T, B, C인 ABC 암호화 스킴을 (B, C, S, T)-SMES로, $T = T_1 \otimes T_2$인 TSMES를 (B, C, S, T_1, T_2)-TSMES로 간편하게 지칭하였다. 이 논문에서 저자는 (B, C, S, T_1, T_2)-TSMES가 $T = I_n$인 (B', C', S', I_n)-SMES와 동치임을 보여, TSMES는 $T = I_n$인 SMES와 같은 안전성을 가지기 때문에 원래의 SMES보다 안전성이 떨어진다고 주장하였다.

정리 1. $B, C \in F_q[x_1, \cdots, x_n]^{n \times n}$, $S \in F_q^{m \times m}$, $T_1, T_2 \in F_q^{s \times s}$라고 하자. (B, C, S, T_1, T_2)-TSMES은 적당한 $B', C' \in F_q[x_1, \cdots, x_n]^{n \times n}$, $S' \in F_q^{m \times m}$, 그리고 F_q상의 $n \times n$ 항등행렬 I_n에 대해 (B', C', S', I_n)-SMES와 동치이다.

증명. 위에서 이미 언급한 것처럼 (B, C, S, T)-SMES의 공개키는 아래의 형태로 표기된다.

$$P(M) = S\psi(\varphi(TM)\varphi(BTM), \varphi(TM)\varphi(CTM))$$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

그런데 $T = T_1 \otimes T_2$이므로 위의 보조정리 2의 증명에서와 같이 $\varphi(TM) = \varphi((T_1 \otimes T_2)M) = T_1\varphi(M)T_2^{\mathrm{T}}$가 성립하므로 아래의 식이 성립한다.

$$P(M) = S\psi(T_1\varphi(M)T_2^{\mathrm{T}}\varphi(BTM), T_1\varphi(M)T_2^{\mathrm{T}}\varphi(CTM))$$

한편 임의의 $y \in F_q^n$에 대해 $T_2^{\mathrm{T}}\varphi(y) = \varphi((T_2^{\mathrm{T}} \otimes I_s)y)$가 성립하고, 임의의 $L_1, L_2 \in F_q^{s \times s}$에 대해 $\psi(T_1L_1, T_1L_2) = (T_1 \otimes I_{2s})\psi(L_1, L_2)$가 성립하므로,

$$P(M) = S\psi(T_1\varphi(M)\varphi((T_2^t \otimes I_s)BTM), T_1\varphi(M)\varphi((T_2^t \otimes I_s)CTM))$$

$$= S(T_1 \otimes I_{2s})\psi(\varphi(M)\varphi((T_2^t \otimes I_s)BTM), \varphi(M)\varphi((T_2^t \otimes I_s)CTM)).$$

이제, $B' = (T_2^t \otimes I_s)BT$, $C' = (T_2^t \otimes I_s)CT$, 그리고 $S' = S(T_1 \otimes I_{2s})$로 잡으면,

$$P(M) = S'\psi(\varphi(M)\varphi(B'M), \varphi(M)\varphi(C'M))$$

을 만족한다. 이는 (B', C', S', I_n)-SMES의 공개키의 표현이므로, 이로써 (B, C, S, T_1, T_2)-TSMES는 (B', C', S', I_n)-SMES와 동치임이 증명되었다. ■

따라서, TSMES는 기존의 SMES보다 훨씬 취약하며, 복호화 실패 문제는 여전히 문제점으로 남아있다.

제 3 절. 신원확인 프로토콜로부터 전자서명의 설계

대부분의 MQ-기반 전자서명들은 MQ 문제뿐만 아니라 IP 문제에도 기반하고 있다. 이러한 연구방향은 효율적인 MQ기반 암호시스템을 가능하게 하지만, 그 안전성 증명에는 어려움을 주고 있는 것이 사실이다. 이러한 흐름과는 다르게 MQ 문제에만 기반하여 설계된 신원확인(identification) 프로토콜이 있다. 이것은 그 안전성을 NP complete 문제라고 알려진 MQ 문제로 귀결시킬 수 있어서 안전성 증명이 가능하다는 장점이 있다. 또한 이것으로부터 만들어진 전자서명도 역시 이러한 안전성 증명이 가능하다. 이것의 단점은 IP 문제에 기반한 암호시스템들보다 효율성이 떨어진다는 점이었는데 최근에 컴퓨터 하드웨어 기술의 발전으로 이러한 단점도 많이 극복한 시스템들이 제안되고 있다. 이 절에서는 MQ 문제에 기반한 신원확인(identification) 프로토콜과 이것으로부터 만들어진 전자서명에 대해서 살펴보겠다.

1. MQ 문제에 기반한 신원확인(identification) 프로토콜

Crypto 2011에서 Sakumoto 등은 2가지 새로운 신원확인 스킴, 3-pass 신원확인,

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

5-pass 신원확인 스킴을 제안하였다 [SSH11]. 먼저 F_q를 q개의 원소를 가지는 유한체, $x=(x_1,.,,,x_n)$라 하고, F: $F_q^n \to F_q^m$를 이차다변수 함수라 하자. 이 때, 다음과 같은 집합을 정의한다.

$$MQ(n,m,F_q)=\{\mathrm{F}(x)=(f_1(x),f_2(x),...,f_m(x)) \mid f_s(x)=\sum a_{ij}^{(s)}x_ix_j+\sum b_i^{(s)}x_i,\ 1\le s\le m\ \}.$$

또한, G(x,y)=F(x+y)−F(x)−F(y)를 함수 F의 polar 형태라 부른다. MQ 문제 MQ(F,v)는 다음과 같이 정의된다.

정의 3(MQ 문제 MQ(F,v)). 주어진 벡터 $v\in F_q^m$에 대해서 F(s)=v가 되는 벡터 $s\in F_q^n$가 존재한다면 s를 찾아라.

이 문제의 결정버전(decision version)은 1979년에 NP complete 임이 밝혀졌고, $MQ(n,m,F_q)$에서 임의로 선택된 F와 F_q^n에서 임의로 선택된 s에 대해서 $v=\mathrm{F}(s)$가 주어져 있을 때, MQ(F,v) 문제의 해를 무시할 수 없는 확률로 찾는 확률적 다항식 시간의 알고리즘은 존재하지 않을 것으로 여겨지고 있다. Sakumoto 등은 함수 F의 polar 형태를 이용하여 비밀정보를 나누는 방법을 이용하여 MQ 문제 MQ(F,v)와 commitment 스킴의 안전성에만 기반한 신원확인 스킴을 제안하였다. 구체적으로 비밀 $s=r_1+r_2$로 나눴다고 하면 공개값 $v=\mathrm{F}(s)$는 다음과 같이 표현될 수 있다.

$$v=\mathrm{F}(r_0)+\mathrm{F}(r_1)+\mathrm{G}(r_0,r_1)$$

r_0와 F(r_0)는 다음과 같이 더욱 나눠진다.

$$\alpha r_0=t_0+t_1 \text{ and } \alpha\mathrm{F}(r_0)=e_0+e_1$$

그러면 폴라폼 G의 선형성에 의해서 다음이 성립함을 알 수 있다.

$$\alpha v=(e_1+\alpha\mathrm{F}(r_1)+\mathrm{G}(t_1,r_1))+(e_0+\mathrm{G}(t_0,r_1))$$

여기에서 αv을 구성하는 각 덧셈의 부분 $(e_1+\alpha\mathrm{F}(r_1)+\mathrm{G}(t_1,r_1))$와 $(e_0+\mathrm{G}(t_0,r_1))$는 각각 (r_1,t_1,e_1)와 (r_1,t_0,e_0)에만 의존함을 알 수 있고, 이것으로부터 비밀 s에 대한 정보는 아무것도 드러나지 않게 된다. 공개키를 $v(=\mathrm{F}(s))$, 비밀키를 s라 할 때, 신원확인 스킴은 그림 1과 같다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

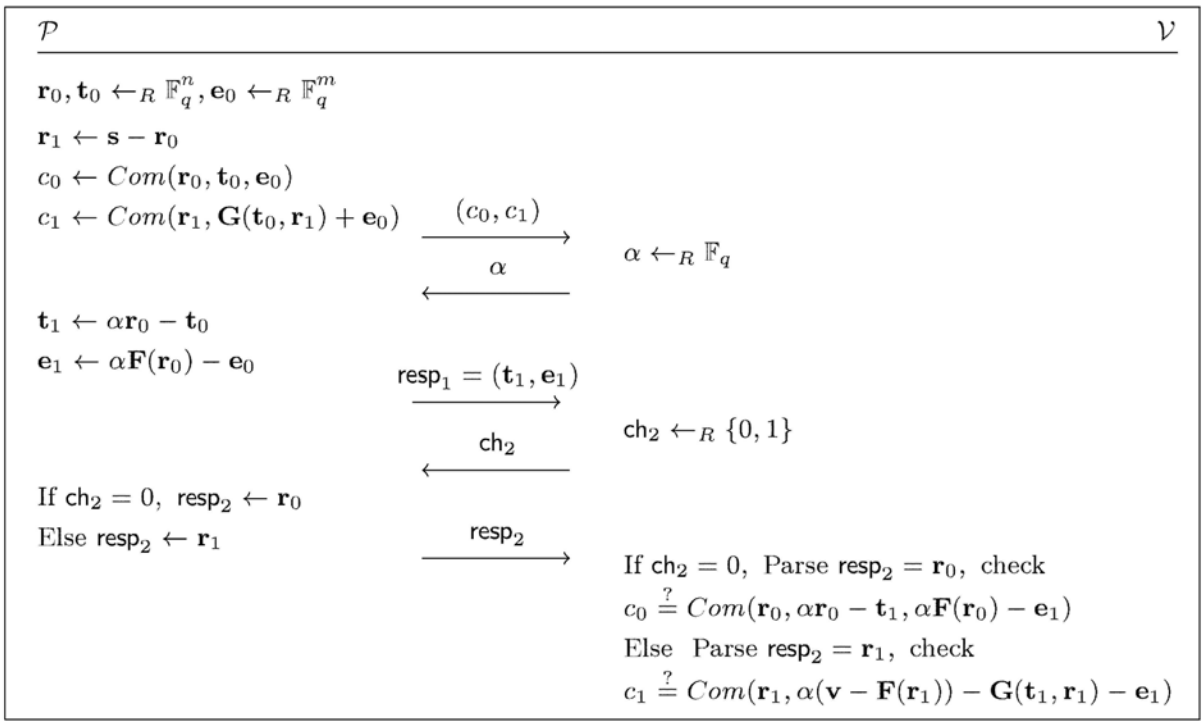


그림 1. 5-pass 신원확인 스킴

이 스킴의 correctness는 다음과 같다.

- $\alpha r_0 - t_1 = \alpha r_0 - (\alpha r_0 - t_0) = t_0$
- $\alpha \mathrm{F}(r_0) - e_1 = \alpha \mathrm{F}(r_0) - (\alpha \mathrm{F}(r_0) - e_0) = e_0$
- $$\begin{aligned} \mathrm{G}(t_1, r_1) &= \mathrm{G}(\alpha r_0 - t_0, r_1) \\ &= \mathrm{G}(\alpha r_0, r_1) - \mathrm{G}(t_0, r_1) \\ &= \alpha \mathrm{G}(r_0, r_1) - \mathrm{G}(t_0, r_1) \\ &= \alpha \mathrm{F}(r_{0+r_1}) - \alpha \mathrm{F}(r_{0)-\alpha} \mathrm{F}(r_1) - \mathrm{G}(t_0, r_1) \\ &= \alpha \mathrm{F}(s) - (e_0 + e_1) - \alpha \mathrm{F}(r_1) - \mathrm{G}(t_0, r_1) \end{aligned}$$
- $\alpha(\mathrm{v} - \mathrm{F}(r_1)) - \mathrm{G}(\mathrm{t}_1, \mathrm{r}_1) - \mathrm{e}_1 = \mathrm{G}(\mathrm{t}_0, \mathrm{r}_1) + \mathrm{e}_0$

정리 2. 5-pass 신원확인 프로토콜은 commitment 스킴 Com이 계산적으로 binding 되었을 때, $\frac{q+1}{2q}$의 오류를 가지고 건전성을 가진다.

2. 신원확인(identification) 프로토콜로부터 전자서명 설계

신원확인 스킴이 주어져 있을 때, 이것을 전자서명으로 전환시키는 몇 가지 방법들이 알려져 있다. 여기에서는 위에서 소개한 5-pass 신원확인 스킴을 Fiat-Shamir 변환방법

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

을 이용하여 전자서명으로 전환한 Chen 등의 결과 [CHRS16]를 살펴보겠다. 여기에서는 앞에서 소개한 신원확인 프로토콜과 같이 두 challenge의 길이가 각각 q와 2로 제한된 스킴만을 고려할 것이다.

정의 4. (q2-신원확인 스킴). k를 자연수라 할 때, q2-신원확인 프로토콜 IDS(1^k)는 challenge 공간 C_1, C_2가 $|C_1|=q$, $|C_2|=2$를 만족하는 5-pass 신원확인 프로토콜이다. 더욱이 commitment com이 주어진 값을 취한 확률은 k에 대해서 무시가능하다.

정의 5. (q2-Extractor). q2-신원확인 스킴 IDS(1^k)가 다음을 만족할 때, q2-Extractor ε를 가진다고 정의한다. 공개키 pk와 다음 조건

$$ch_1^{(1)} = ch_1^{(2)} \neq ch_1^{(3)} = ch_1^{(4)}$$
$$ch_2^{(1)} = ch_2^{(3)} \neq ch_2^{(2)} = ch_2^{(4)}$$

을 만족하는 4개의 transcripts $trans^{(i)} = (com, ch_1^{(i)}, resp_1^{(i)}, ch_2^{(i)}, resp_2^{(i)})$ $i \in \{1,2,3,4\}$가 주어져 있을 때, 무시할 수 없는 확률로 pk에 대응하는 비밀키 sk를 내놓는 확률적 다항식 시간 알고리즘 ε가 존재한다.

$IDS^r = (KGen, P^r, V^r)$을 신원확인 스킴 $IDS = (KGen, P, V)$의 r라운드의 평행한 합성이라고 하자. 또한 j 라운드의 transcript를 $trans_j = (com_j, ch_{1,j}, resp_{1,j}, ch_{2,j}, resp_{2,j})$로 표기한다. 그러면 q2-신원확인 스킴의 Fiat-Shamir 변환은 표 10과 같다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

q2-신원확인 스킴의 Fiat-Shamir 변환

1: 자연수 k는 안전도 파라미터, $IDS=(KGen,P,V)$는 건전성 오류 κ를 가지는 q2 신원확인 스킴이라 하자.

다음을 만족하는 신원확인 스킴의 라운스 수 r을 선택한다.

2: $$\kappa^r = \mathrm{negl(k)}$$

challenge 공간 IDS^r, C_1^r, C_2^r이 k에 대해서 지수의 크기를 가진다.

암호학적 해쉬함수 H_1, H_2를 선택한다.

3: $$H_1 : \{0,1\}^* \rightarrow C_1^r \text{와 } H_2 : \{0,1\}^* \rightarrow C_2^r$$

IDS로부터 변환된 q2 전자서명 $q2-Dss(1^k)$는 다음을 만족하는 알고리즘 $(KGen, Sign, Vf)$이다.

- $(sk,pk) \leftarrow KGen(1^k)$
- $\sigma=(\sigma_0,\sigma_1,\sigma_2) \leftarrow Sign(sk,m)$ 여기에서

4: $$\begin{aligned} \sigma_0 &= com \leftarrow P_0^{r(sk)}, \\ h_1 &= H_1(m,\sigma_0), \\ \sigma_1 &= resp_1 \leftarrow P_1^r(sk,\sigma_0,h_1) \\ h_2 &= H_2(m,\sigma_0,h_1,\sigma_1), \\ \sigma_2 &= resp_2 \leftarrow P_2^r(sk,\sigma_0,h_1,\sigma_1,h_2) \end{aligned}$$

- $Vf(pk,m,\sigma=(\sigma_0,\sigma_1,\sigma_2))$: 해쉬값 $h_1=H_1(m,\sigma_0)$, $h_2=H_2(m,\sigma_0,h_1,\sigma_1)$을 계산하고 $V^r(pk,\sigma_0,h_1,\sigma_1,h_2,\sigma_2)$를 내놓는다.

표 10. q2-신원확인 스킴의 Fiat-Shamir 변환

정리 3. (q2-전자서명의 EU-CMA 안전성) 자연수 k에 대해서 정직한 verifier 영지식인 q2 신원확인 프로토콜 IDS(1^k)가 상수의 건전성 오류 κ와 q2 extractor를 가진다고 하자. 그러면 Fiat-Shamir 변환을 통한 q2 전자서명 $q2-DSS(1^k)$는 adaptive하게 선택된 평문 공격에 대해서 existentially 위조불가능하다.

이 정리의 증명을 위해서는 몇 가지 보조정리가 필요하다. 지금부터 해쉬함수 H_1, H_2를 독립적인 랜덤오라클 O_1, O_2로 모델링할 것이다.

보조정리 3. (q2-전자서명의 forking lemma) 만약 주어진 공개키에 대해서 무시할 수 없는 확률로 유효한 전자서명과 평문 쌍 (m,σ)을 만들 수 있는 확률적 다항식 시간의 공격자 A가 있다면 같은 랜덤과 다른 오라클들을 가지고 A를 되돌리는 것에 의해서 다음

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

이 성립하는 해쉬값을 가지는 4개의 유효한 전자서명을 무시할 수 없는 확률로 만들어 낼 수 있다.

$$h_{1,j}^{(1)} = h_{1,j}^{(2)} \neq h_{1,j}^{(3)} = h_{1,j}^{(4)}$$
$$h_{2,j}^{(1)} = h_{2,j}^{(3)} \neq h_{2,j}^{(2)} = h_{2,j}^{(4)}$$

보조정리 4. 자연수 k에 대해서 IDS(1^k)를 정직한 verifier 영지식인 q2 신원확인 스킴이라고 하자. 그러면 Fiat-Shamir 변환에 의해서 얻어진 q2 전자서명 $q2-DSS(1^k)$의 EU-CMA 안전성에 대한 확률적 다항식 시간의 공격자 B는 q2-전자서명의 forking lemma의 성질을 가지는 key-only 공격자 A로 전환될 수 있다. A는 본질적으로 B와 같은 성공확률을 가지고 다항식 시간안에 성공한다.

(**q2-전자서명의 EU-CMA 안전성 정리의 증명**) 무시할 수 없는 확률로 성공하는 $q2-DSS(1^k)$의 EU-CMA 안전성에 대한 확률적 다항식 시간의 공격자 B가 존재한다고 가정하자. 그러면 위의 보조정리를 적용하여 신원확인 스킴의 건전성을 깨는 확률적 다항식 시간의 impersonator C는 본질적으로 B와 같은 성공확률을 가지는 확률적 다항식 시간의 key-only 공격자 A를 만들 수 있다. 주어진 신원확인 스킴의 공개키에 대해서 C는 q2-전자서명의 forking lemma에서와 같이 A를 실행할 수 있다. 즉, C가 q2-extractor ε에서 요구되는 4개의 transcript를 이끄는 4개의 전자서명을 얻기 위해서 A를 사용할 수 있다. 그러면 ε를 실행해서 C는 성공확률 1로 P를 가장하게 하는 유효한 비밀키를 얻어 낼 수 있다. C는 두 개의 확률적 다항식 시간의 A와 ε를 실행하였으므로 무시할 수 없는 확률로 다항식 시간안에 성공할 수 있다.

3. MQDSS 전자서명

위에서 설명한 q2-전자서명은 유한체 F_{31}위에서 구체화 되었고, 이를 MQDSS라고 부른다 [CHRS16]. 먼저 MQDSS의 파라미터는 다음과 같다.

파라미터: 안전도 파라미터 k에 대하여 $MQ(n,m,F_{31}) \geq k$를 만족하는 자연수 m,n을 선택한다. 그리고 F의 총 계수의 개수 $F_{len} = m \cdot \frac{n(n+1)}{2}$라 놓는다. 다음과 같은 함수가 필요하다.

- 암호학적 해쉬함수 $H: \{0,1\}^* \rightarrow \{0,1\}^k$, $H_1 : \{0,1\}^{2k} \rightarrow F_{31}^r$, $H_2 : \{0,1\}^{2k} \rightarrow \{0,1\}^r$
- 두 개의 스트링 commitment 함수
  $Com_0 : F_{31}^n \times F_{31}^n \times F_{31}^m \rightarrow \{0,1\}^k$, $Com_1 : F_{31}^n \times F_{31}^m \rightarrow \{0,1\}^k$
- 의사난수생성기 $G_{S_F} : \{0,1\}^k \rightarrow F_{31}^{F_{len}}$, $G_{SK} : \{0,1\}^k \rightarrow F_{31}^n$, $G_c : \{0,1\}^{2k} \rightarrow F_{31}^{r(2n+m)}$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

키생성: 주어진 안전도 파라미터 k에 대해서

- k비트 비밀키 SK를 랜덤하게 선택한다. 또한 시드 S_F도 랜덤하게 선택한다.
- 시드 S_F를 확장하는 것에 의해서 $MQ(n,m,F_{31})$로부터 의사난수 MQ 시스템 F를 생성한다.
- $SK_{F_{31}} = G_{SK}(SK)$를 계산한다.
- $PK_v = \mathrm{F}(\mathrm{SK}_{\mathrm{F}_{31}})$을 계산한다.
- 비밀키 sk=(SK, S_F)라 놓고, pk=(S_F, PK_v)라 놓는다

서명생성: 평문 $m \in \{0,1\}^*$와 비밀키 sk=(SK, S_F)를 가지고 실행된다.

- $\mathrm{F} = \mathrm{G}_{\mathrm{S_F}}(\mathrm{SK_F})$을 계산한다.
- $R = H(SK \| m)$와 $D = H(R \| m)$을 계산한다.
- $G_c(SK, D) \rightarrow (r_{(0,0)}, ..., r_{(0,r-1)}, t_{(0,0)}, ..., t_{(0,r-1)}, e_{(0,0)}, ..., e_{(0,r-1)})$를 계산한다.
- $c_{(0,i)} = Com_0(r_{(0,i)}, t_{(0,i)}, e_{(0,i)})$와 $c_{(1,i)} = Com_1(r_{(1,i)}, \mathrm{G}(\mathrm{t_{(0,i)}}, \mathrm{r_{(1,i)}}) + \mathrm{e_{(0,i)}})$를 계산한다. 여기에서 $r_{(1,i)} = SK - r_{(0,i)}$.
- $\sigma_0 = H(c_{(0,0)} \| c_{(1,0)} \| ... c_{(0,r-1)} \| c_{(1,r-1)})$를 계산한다.
- $h_1 = (D, \sigma_0)$, $H_1(h_1) = (\alpha_1, ..., \alpha_r)$를 계산한다.
- $t_{(1,i)} = \alpha_i r_{(0,i)} - t_{(0,i)}$, $e_{(1,i)} = \alpha_i \mathrm{F}(\mathrm{r_{(0,i)}}) - \mathrm{e_{(0,i)}}$를 계산한다.
- $\sigma_1 = (t_{(1,0)} \| e_{(1,0)} \| ... t_{(1,r-1)} \| e_{(1,r-1)})$라 놓는다.
- $h_2 = H_2(D, \sigma_0, h_1, \sigma_1) = (ch_{2,0}, ..., ch_{2,r-1})$를 계산한다.
- $\sigma_2 = (r_{(ch_{2,0},0)}, ..., r_{(ch_{2,r-1},r-1)}, c_{1-ch_{2,0}}, ..., c_{1-ch_{2,r-1}})$라 놓는다.
- 그러면 서명 $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$이 된다.

서명검증: 평문 $m \in \{0,1\}^*$, 서명 $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$와 공개키 pk=$(S_F, PK_v)$를 가지고 실행된다.

- $D = H(R \| m)$, $h_1 = (D, \sigma_0)$, $H_1(h_1) = (\alpha_1, ..., \alpha_r) = \alpha$를 계산한다.
- $h_2 = H_2(D, \sigma_0, h_1, \sigma_1) = (ch_{2,0}, ..., ch_{2,r-1})$를 계산한다.
- $ch_{2,i} (i = 0, ..., r-1)$의 값에 따라 다음을 계산한다.

만약 $ch_{2,i} = 0$이면, $c_{(0,i)} = Com_0(r_i, \alpha_i r_i - t_i, \alpha_i \mathrm{F}(\mathrm{r_{(0,i)}}) - \mathrm{e_i})$를 계산하고,

만약 $ch_{2,i} = 1$이면, $c_{(1,i)} = Com_1(r_i, \alpha_i (PK_v - \mathrm{F}(\mathrm{r_i})) - \mathrm{G}(\mathrm{t_i}, \mathrm{r_i}) - \mathrm{e_i})$를 계산한다.

- $\sigma'_0 = H(c_{(0,0)} \| c_{(1,0)} \| ... c_{(0,r-1)} \| c_{(1,r-1)})$를 계산한다.
- 만약 $\sigma'_0 = \sigma_0$를 만족하면 서명확인하고 그렇지 않으면 올바르지 않은 서명으로 확인한다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

MQDSS 전자서명은 F_{31}의 한 원소를 5비트를 사용하여 저장한다고 가정하면 표 11과 같은 크기를 가진다.

비밀키 sk	공개키 pk	서명 σ
$2k$ 비트	$k+5m$ 비트	$(2+r)k+5r(2n+m)$ 비트

표 11. MQDSS 전자서명의 크기

MQDSS 전자서명의 안전한 파라미터 선택을 위해서 $MQ(n,m,F_{31})$에서 MQ 문제를 해결하는 Gröbner 알고리즘의 변형인 F5의 Hybrid 알고리즘의 계산복잡도를 고려하였다. 이것은 다음과 같이 주어진다.

$$C_{hyb}(n,m,\tau,d_{reg}(n(1-\tau),m)) = q^{\tau n} \cdot C_{F5}(n(1-\tau),m,d_{reg}(n(1-\tau),m))$$

여기에서 τ는 0과 1사이의 상수이고 d_{reg}와 C_{F5}는 다음과 같이 주어진다.

$$d_{reg}(n,m) \approx (\frac{m}{n} - \frac{1}{2} - \sqrt{\frac{m}{n}(\frac{m}{n}-1)}) + O(n^{1.3})$$

$$C_{F5}(n,m) = O\left(\left(m\binom{n+d_{reg}(n,m)-1}{d_{reg}(n,m)}\right)^{w}\right)$$

여기에서 w은 선형대수 상수로 2와 3사이의 수이다. 이러한 계산복잡도로부터 MQ 문제의 복잡도가 2^{128}을 가지기 위해서는 변수의 개수 n은 51보다 커야 한다는 것을 알 수 있다. 또한 Hybrid 알고리즘은 $m > n$인 경우 더 빠른 계산속도를 보여준다는 것이 알려져 있고, $m < n$인 경우 몇 가지 변수를 추측하는 것에 의해서 $m = n$인 경우로 만들 수 있기 때문에 $m = n$으로 선택하는 것이 좋다. 또한 MQDSS 전자서명의 구현 효율성을 위해서 51보다 큰 16의 배수인 64를 선택하는 것이 효율적이므로 $m = n = 64$를 선택하는 것이 128비트의 안전도와 효율성을 주는 파라미터라 할 수 있다.

양자알고리즘에 대한 안전도를 고려한 파라미터의 선택을 위해서는 아직까지 MQ 문제를 해결하는 양자알고리즘이 알려져 있지 않지만, MQ 문제를 해결하기 위해 Grover의 검색 알고리즘 [G96]을 적용하는 것과 Hybrid 알고리즘에 Grover의 검색 알고리즘을 접목시키는 방법을 고려해 볼 수 있다. Grover의 검색 알고리즘은 크기 N을 가지는 아이템들의 리스트에서 원하는 조건을 만족하는 M개의 아이템을 찾는데 $O(\sqrt{N/M})$의 계산복잡도를 가진다. 이것을 MQ 문제, 즉 F(x) = v를 만족하는 $x \in F_{31}^{n}$를 찾는 것에 적용을 하면, 일단 $N = n\log 31$임을 알 수 있다. 또한 n개의 변수와 n개의 식을 가지는 연립방정식의 해의 개수는 $\lambda = 1$을 가지는 포아송 분포를 따른다는 것이 알려져 있기 때문에 F(x) = v가 최소한 M개의 식을 확률은 $P[X \geq M] \geq \frac{1}{e}(\frac{e}{M})^{M}$임 됨을 알 수 있다. 따라서 M이 5이상일 확률도 이미 2^{-8}보다 크기 때문에 M을 4이하라 가정하여도 무리가 없다. 따라서 MQ 문제 F(x) = v를 푸는 Grover 알고리즘의 복잡도는 $O(2^{n\log 31/2}/4) \approx 2^{156}$임을

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

알 수 있다.

MQDSS는 128비트의 안전성을 위해서 F_{31}위에서 라운드 수 $r=269$와 F의 변수의 개수와 식의 개수는 모두 64인 파라미터를 제시하였다. 또한 M을 4이하이고 n=m인 경우에 Hybrid 알고리즘의 복잡도도 약 2^{139}정도가 됨을 알 수 있다. 마지막으로 MQDSS의 128비트 포스트 양자 안전도에 대한 EU-CMA를 달성하기 위해서 양자알고리즘의 공격자가 challenge들을 효과적으로 제어하기 위한 preimage 검색을 수행할 수 있기 때문에 이를 막기 위해서 $\kappa^r \leq 2^{-256}$을 만족해야 한다. 그런데 신원확인 스킴에서 $\kappa = \frac{q+1}{2q}$를 만족해야 했기 때문에 $q=31$에 대해서 $r=269$정도가 되어야 함을 알 수 있다. 따라서 128비트 포스트 양자 안전도를 위해서는 MQDSS의 파라미터로 $q=31$, $r=269$, $m=n=64$를 선택해야 됨을 알 수 있다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 3 장. 연구 수행 내용 및 성과

제 1 절. 다변수 이차식 기반 전자서명 알고리즘 안전성 분석

1. RGB 안전성 분석

RGB 전자서명 알고리즘은 2015년에 Shen과 Tang에 의해 발표된 다변수 다항식 기반 서명 알고리즘이다[ST15]. 저자들은 연산량이나 저장 공간을 적게 요구하여 경량 기기에 사용할만한 서명 알고리즘으로 RGB를 추천하였다. 그들은 RGB를 경량기기에 구현한 결과를 국제학회에 발표하여[TLS16], RGB를 경량기기의 보안을 위해 충분히 사용가능하다고 주장하였다.

이제 우리는 이렇게 구현된 RGB 서명 알고리즘이 저자들의 주장과는 달리, 공격자가 서명 값을 원하는 메시지의 유효한 서명 값을 쉽게 찾을 수 있으며 따라서 안전하지 않다는 것을 보일 것이다. 저자들이 구현한 80비트 RGB 알고리즘의 유효한 서명값을 위조해내는데 1.7초가 걸렸으며, 128비트 RGB 알고리즘의 유효한 서명값을 찾는데 일주일 정도가 걸린다. 이러한 RGB 알고리즘이 저자가 주장하는 안전성을 갖도록 하려면 파라미터를 크게 잡아야 하는데, 이렇게 할 경우 키 길이도 길어질 뿐만 아니라 서명할 때와 서명을 확인할 때 걸리는 시간이 기존의 다변수 다항식 기반 서명 알고리즘보다 길어지기 때문에, 저자들이 주장한대로 경량기기에서 사용할 수 없게 된다.

(1) RGB 전자서명 알고리즘

RGB 전자서명 알고리즘은 대부분의 다변수 다항식 기반 암호의 구조가 그렇듯, 그림 2와 같이 $P = S \circ F \circ T$의 구조를 사용한다. 대부분의 다변수 다항식 기반 암호에서 $S : F_q^m \mapsto F_q^m$, $T : F_q^n \mapsto F_q^n$은 affine 변환 혹은 선형변환이며, $F : F_q^n \mapsto F_q^m$은 쉽게 역연산이 가능한 다변수 2차식 시스템이다. 즉, S, F, T는 쉽게 역연산이 가능하다. 이러한 S, F, T를 비밀키로 하고 $P = S \circ F \circ T$를 공개키로 하며 메시지 M의 해쉬값 $h \in F_q^m$에 대해 $P(\gamma) = h$을 만족하는 $\gamma \in F_q^n$이 메시지 M의 서명값이 된다. 공개키를 아는 사람은 $P(\gamma) = h$이 성립하는지의 여부를 통해 누구든지 서명값 γ가 유효한 서명인지 확인할 수 있다. 비밀키를 아는 사람은 메시지 M에 해당하는 서명값 γ를 찾을 수 있지만, 그렇지 않은 사람은 메시지에 해당하는 서명값을 찾으려면 $P(X) = h$이라는 다변수 다항식 시스템을 풀어야 하는데, 이러한 풀이는 어려운 문제이기 때문에 주어진 시간내에 답을 찾을 수 없다는 것이 다변수 다항식 기반 암호의 핵심 아이디어다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

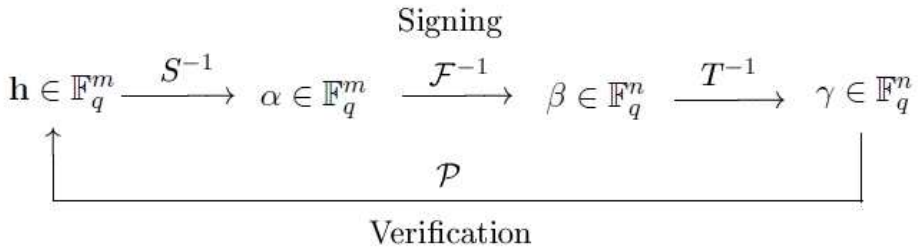


그림 2. 일반적인 다변수 다항식 기반 서명 알고리즘의 구조

하지만 RGB 서명 알고리즘은 이러한 대부분의 다변수 다항식 기반 암호가 갖는 형태와는 다른 "mixed type"이다. "mixed type"에서 역시 공개키와 비밀키는 $P=S\circ F\circ T$의 관계를 만족한다. 하지만, 메시지 M과 서명 값 σ가 $P(\sigma)=h$을 만족하는 대부분의 다변수 다항식 기반 암호와는 달리 $P(\sigma,h)=0$을 만족한다. 이러한 mixed type 서명 알고리즘에서 서명을 생성하기 위해서는 T가 일반적인 affine 변환(혹은 선형 변환)이 아닌 $T(x_1,...,x_n)=(T_1(x_1,...,x_r),T_2(x_{r+1},...,x_n))$처럼 두 개의 affine 변환(혹은 선형 변환)을 연결해놓은 형태이어야 한다. 이러한 T의 형태를 왜 가져야 하는지 서명 과정에서 확인할 수 있는데, 이러한 사실은 우리가 이 RGB 서명 알고리즘을 공격하는데 결정적인 역할을 했다. 이것은 해당 내용이 나오는 뒷부분에서 자세히 언급하도록 하겠다.

RGB 서명 알고리즘에 대해 자세히 언급하기에 앞서 먼저 3색 다항식(Three-color Polynomial)과 3색 함수(Three-color Map)를 정의하도록 한다.

정의 6. (3색 다항식) 어떤 다항식 $f\in F_q[y_1,\cdots,y_r,z_1,\cdots,z_g,t_1,\cdots,t_b]$이 아래와 같은 형태를 가졌을 경우 f를 3색 다항식이라고 한다. (여기서 $\alpha_{i_1i_2},\beta_{ij},\gamma_{ik},\delta_{jk},\epsilon_{k_1k_2},\alpha_i,\beta_j,\gamma_k,C\in F_q$)

$$f=\sum_{1\le i_1\le i_2\le r}\alpha_{i_1i_2}y_{i_1}y_{i_2}+\sum_{i=1}^{r}\sum_{j=1}^{g}\beta_{ij}y_iz_j+\sum_{i=1}^{r}\sum_{k=1}^{b}\gamma_{ik}y_it_k+\sum_{j=1}^{g}\sum_{k=1}^{b}\delta_{jk}z_jt_k+\sum_{1\le k_1\le k_2\le b}\epsilon_{k_1k_2}t_{k_1}t_{k_2}$$
$$+\sum_{i=1}^{r}\alpha_iy_i+\sum_{j=1}^{g}\beta_jz_j+\sum_{k=1}^{b}\gamma_kt_k+C$$

정의 7. (3색 함수) 어떤 함수 $F\colon F_q^{r+g+b}\mapsto F_q^g$, $F(y_1,\cdots,y_r,z_1,\cdots,z_g,t_1,\cdots,t_b)=(f_1,\cdots,f_g)$의 각 성분함수 $f_i(1\le i\le g)$가 정의 6에서 정의된 3색 다항식일 경우 F를 3색 함수라고 한다.

위에서 정의된 개념들은 특별한 수학적 의미를 갖고 있지는 않다. 그저 저자들이 자신들의 알고리즘을 좀 더 쉽게 설명하기 위해서 만든 개념이다. 여기서 3색의 개념을 사용했듯, 서명 알고리즘의 이름에도 3색의 개념을 이용하여 빛의 3원색인 Red-Green-Blue를 따 RGB라고 한 것이다. 3색 다항식을 보면 뒤의 서명 알고리즘에서 메시지 값이 들어갈 변수 부분을 $y_i(1\le i\le r)$로 표기하였고, 나머지 서명 값을 구하기 위한 변수들에는

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Oil-Vinegar 서명 알고리즘에서 사용한 구조(앞으로 이를 Oil-Vinegar 구조라고 부르기로 한다.)를 쥐서 $z_{j_1}z_{j_2}(1 \le j_1, j_2 \le g)$부분이 나타나지 않았다. UOV(Unbalanced Oil and Vinegar) 서명 알고리즘[KPG99]을 아는 사람들의 이해를 돕기 위해 z_j, t_k의 변수들을 모두 $x_j(1 \le j \le g+b)$의 형태로 표기하도록 하겠다. 이 경우 비밀함수 F의 각 항의 모양은 아래와 같다.

$$f = \sum_{1 \le i_1 \le i_2 \le r} \alpha_{i_1 i_2} y_{i_1} y_{i_2} + \sum_{i=1}^{r}\sum_{j=1}^{g+b} \beta_{ij} y_i x_j + \sum_{j_1=1}^{g}\sum_{j_2=g+1}^{g+b} \gamma_{j_1 j_2} x_{j_1} x_{j_2} + \sum_{g+1 \le j_1 \le j_2 \le g+b} \delta_{j_1 j_2} x_{j_1} x_{j_2} + \sum_{i=1}^{r} \alpha_i y_i + \sum_{j=1}^{g+b} \beta_j x_j + C$$

S의 경우 임의의 선형 변환이지만, T는 앞에서 설명한 것처럼 두 선형변환을 연결한 것이다. 두 affine 변환 $T_1 : F_q^r \mapsto F_q^r$, $T_2 : F_q^{g+b} \mapsto F_q^{g+b}$는 아래와 같이 정의된다.

$$T(y_1, \cdots, y_r, x_1, \cdots, x_{g+b}) = (T_1(y_1, \cdots, y_r) \| T_2(x_1, \cdots, x_{g+b}))$$

위와 같은 정의를 바탕으로 RGB의 키 생성, 서명 생성, 서명 검증의 방법에 대해 자세한 설명은 다음과 같다.

■ 키 생성 방법

① 임의의 선형 변환 $S: F_q^g \mapsto F_q^g$를 선택하고, 2차식 F와 affine 변환 $T: F_q^n \mapsto F_q^n$를 위에서 설명한 형태의 함수가 되도록 선택한다.

② $P = S \circ F \circ T$를 계산한다.

③ P가 공개키, S, F, T는 비밀키이다.

■ 서명 생성 방법

서명하고자 하는 메시지 M의 해쉬값 $h = H(M) = (h_1, \cdots, h_r) \in F_q^r$에 대응하는 서명값 $\sigma = (\sigma_1, \cdots, \sigma_g, \sigma_{g+1}, \cdots, \sigma_{g+b}) \in F_q^{g+b}$을 계산하는 방법은 다음과 같다.

① $(t_1, \cdots, t_r) = T_1(h_1, \cdots h_r)$를 계산한다.

② 임의로 b개의 값 $(s_{g+1}, \cdots, s_{g+b}) \in F_q^b$를 선택한다.

③ $t_1, \cdots, t_r, s_{g+1}, \cdots, s_{g+b}$를 F의 각 성분의 $y_1, \cdots, y_r, x_{g+1}, \cdots, x_{g+b}$에 대입한다. 그리하여, 아래 형태의 변수가 $x_1, \cdots, x_g$이고 식이 g개인 1차 연립방정식 시스템을 얻는다.

$$\begin{cases} f_1(t_1, \cdots, t_r, x_1, \cdots, x_g, s_{g+1}, \cdots, s_{g+b}) = 0 \\ \vdots \qquad\qquad \vdots \\ f_g(t_1, \cdots, t_r, x_1, \cdots, x_g, s_{g+1}, \cdots, s_{g+b}) = 0 \end{cases}$$

④ 위에서 얻은 1차 연립방정식 시스템은 식과 변수의 개수가 g개로 동일하므로 가우스 소거법으로 이를 푼다. 만약 이 연립방정식 시스템의 해가 존재하지 않을 경우 ②로 되돌아가 $(s_{g+1}, \cdots, s_{g+b})$를 다시 선택한 후 해가 존재할 때까지 이 과정을 반복

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

한다. 여기서 얻은 해를 $(x_1, \cdots, x_g) = (s_1, \cdots, s_g)$라고 한다.

⑤ $\sigma = (\sigma_1, \cdots, \sigma_{g+b}) = T_2^{-1}(s_1, \cdots, s_g, s_{g+1}, \cdots, s_b) \in F_q^{g+b}$를 계산한다. σ 값이 우리가 원하는 메시지 M의 서명값이다.

■ 서명 검증

서명값 $\sigma = (\sigma_1, \cdots, \sigma_g, \sigma_{g+1}, \cdots, \sigma_{g+b}) \in F_q^{g+b}$이 메시지 M의 해쉬값 $h = H(M) = (h_1, \cdots, h_r) \in F_q^r$에 대응하는 서명값이 맞는지를 확인하기 위해서는 $P(h, \sigma) = 0$이 만족하는지를 확인한다. 등호가 성립하면 유효한 서명이고, 그렇지 않다면 유효하지 않은 서명이다.

위의 방법대로 서명을 생성하면 왜 유효한 것인지를 살펴보자. 서명 생성 과정의 ⑤에서 $T_2(\sigma_1, \cdots, \sigma_{g+b}) = (s_1, \cdots, s_g, s_{g+1}, \cdots, s_{g+b})$임을 알 수 있으므로, ①~④까지 살펴보면,

$$\begin{aligned} 0 &= F(t_1, \cdots, t_r, s_1, \cdots, s_g, s_{g+1}, \cdots s_{g+b}) = F((t_1, \cdots, t_r) \| (s_1, \cdots, s_g, s_{g+1}, \cdots s_{g+b})) \\ &= F(T_1(h_1, \cdots, h_r) \| T_2(\sigma_1, \cdots, \sigma_{g+b})) = F(T(h, \sigma)) = (F \circ T)(h, \sigma) \end{aligned}$$

S가 선형 변환이므로 양쪽의 왼쪽에 취해주면, 아래의 식이 성립하므로, 이 서명과 서명의 확인 과정은 유효하다고 할 수 있다.

$$S(0) = 0 = S((F \circ T)(h, \sigma)) = (S \circ F \circ T)(h, \sigma) = P(h, \sigma)$$

RGB 서명 알고리즘의 서명 생성 과정을 보면 UOV의 서명 생성과정과 굉장히 유사하다는 것을 알 수 있다. 사실 "mixed type UOV"라고 해도 될 정도이다. 하지만 UOV의 경우 T가 일반적인 affine 변환이지만, RGB 서명 알고리즘에서는 T가 두 개의 affine 변환을 연결한 모양이다. T의 모양에서 UOV와 상당한 차이를 보이는데, RGB의 공개키가 mixed type이기 때문에 T는 반드시 그런 모양을 가져야 한다. 먼저 서명 생성 과정의 ①에서 만약 T_1이 $T_1 : F_q^r \mapsto F_q^r$의 모양이 아니고 T_1의 각 성분 함수가 $(y_1, \cdots, y_r)$ 이외의 다른 변수를 포함하고 있다면, ①에서 어떠한 σ_i 값을 알아야 하는 셈이다. σ_i값은 서명을 생성하는 사람이 찾아야 하는 값이므로, ①의 과정을 진행 중일 때에는 모르고 있는 값이다. 따라서 T_1은 $T_1 : F_q^r \mapsto F_q^r$의 모양을 가져야 한다. 또한 T_2가 만약 언급된 내용이 아니라 $(x_1, \cdots, x_{g+b})$ 이외의 변수를 포함한 항을 가지고 있다면, T_2 자체적으로 가역일 수 없다. 즉, T_2가 가역이려면 반드시 위에서 언급한 모양을 가지고 있어야 한다. 정리하면, T가 두 개의 affine 변환을 연결한 모양이 아니면 서명을 생성할 수 없다.

(2) RGB 서명 알고리즘에 대한 저자들의 안전성 분석

RGB의 저자들은 제안된 서명 알고리즘의 안전성을 보이기 위해 기존의 다변수 다항식

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

기반 암호들을 공격하는데 사용된 공격 방법들을 RGB에 적용하여 이 공격들에 대해 안전하다는 사실을 밝혔다.

(가) UOV attack

이 방법은 (balanced) Oil and Vinegar 서명 알고리즘을 공격할 때 사용되었던 방법이다[KS98, KPS99]. RGB의 경우에도 다변수 2차함수인 F의 모양이 UOV의 그것과 유사하므로 이 공격에 대한 안전성을 분석해야만 한다. 자세한 분석은 해당논문[ST15]를 참고하도록 하고 여기에서는 그 결과만 언급하도록 하겠다. 결과는 3가지의 경우로 나뉜다.

- $r+b=g$일 때 : [KS98]에서 공격당한 OV와 같은 경우로 이 경우에 RGB는 같은 방법으로 공격당할 수 있어 안전하지 않다.
- $r+b \geq \frac{g^2}{2}$일 때 : [KPG99]의 공격을 적용하여 RGB를 공격할 수 있어 안전하지 않다.
- 그 외의 경우 : 공격복잡도는 $q^{r+b-g-1} \cdot g^4$이다. 해당 복잡도가 시스템에서 요구하는 복잡도보다 높아야 한다.

(나) MinRank attack과 Direct Attack

MinRank 공격의 경우 기본적으로 레이어가 1개인 스킴에는 적용되지 않는다. RGB 역시 레이어가 1개인 스킴이므로 이 공격이 적용되지 않는다. 이 경우 이 공격에 대한 복잡도는 $q^{r+b} \cdot g^3$이다.

Direct Attack은 공개키에 있는 다변수 시스템을 풀어 유효한 서명을 얻어내는 공격이다. 메시지에 상관없이 유효한 서명값을 얻으려면, RGB의 경우 식이 g개, 변수가 n개인 다변수 다항식 시스템을 풀어야 한다. 한편, 원하는 메시지에 대한 서명 값을 찾고자 하는 경우 메시지의 해쉬 값을 공개키 함수에 대입한 후 얻어지는 다변수 다항식 시스템을 풀어 서명 값을 찾는 공격을 할 경우, RGB의 경우 메시지 값을 공개키 함수에 대입하면 식이 g개이고 변수의 개수가 $g+b$개인 다변수 다항식 시스템을 얻는다. 하지만, 한 개의 메시지에 대응하는 서명 값은 매우 많기 때문에 이 경우 변수가 식의 개수보다 많은 개수만큼의 변수 값을 추측하여 식과 변수의 개수를 g개로 똑같이 맞춰준 후 다변수 다항식을 풀게 된다. 따라서 이 공격은 g에 의존한다고 할 수 있다. 기본적으로 m개의 식과 n개의 변수를 갖는 다변수 다항식 시스템을 푸는 현재 가장 빠른 알고리즘인 F_5알고리즘의 복잡도는

$$O\left(m\binom{n+d_{reg}-1}{d_{reg}}^{\alpha}\right)$$

이다. 이 때, d_{reg}는 degree of regularity로, 해당 시스템을 F_5로 돌릴 경우 나오는 가장

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

큰 차수를 말한다. 이 값은 Hilbert 시리즈

$$S_{m,n}=\frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-x)^n}(d_i\text{는 해당 다변수 시스템의 } i\text{번째 다항식의 차수})$$

의 양이 아닌 변수가 나오는 가장 작은 차수로 구해진다. 하지만 위의 Hilbert 시리즈를 살펴보면, $m>n$일 때에는 분모를 모두 약분해도 분자에 마이너스 기호가 남아있어 비교적 낮은 수가 나오지만, $m=n$일 경우에는 분모가 모두 약분되고 분자에 양의 계수를 갖는 다항식들의 곱의 형태만 나오기 때문에 $d_{reg}=\sum_{i=1}^{m}d_i-n$이 되는데 이는 꽤 큰 값이다. 그나마 $m<n$일 때는, 분모가 모두 약분되지 않고 남아있으므로, 이 식을 이용하여 d_{reg}를 구할 수도 없다. 그래서 이 경우에는 $n-m$개의 변수를 임의로 추측하여 변수와 식의 개수를 같게 해준 다음에 이 문제를 푼다. 그렇게 식과 변수의 개수가 같다고 해도 위에서 언급한 것처럼 이 경우에도 d_{reg}는 상당히 크기 때문에 복잡도를 낮추기 위해 몇 개의 변수를 추가로 추측하여 시스템을 푸는 Hybrid 방법을 사용한다. 이 방법의 복잡도는 다음과 같다.

$$O\left(\min_{1\le k\le m} q^k\cdot m\binom{n-k+d_{reg}-1}{d_{reg}}^{\alpha}\right)$$

위의 식에서 d_{reg}는 식이 m개이고 변수가 $n-k$개인 다변수 다항식 시스템의 degree of regularity를 말한다.

해당 RGB 서명 알고리즘에서는 $m=n=g$인 상황이다. $2\le\alpha\le3$는 선형 대수학에서 Gauss 소거법을 실행할 때의 복잡도와 연관되는 상수이다. 그런데 RGB의 저자들은 [TW12a]를 인용하여 암호해독자들은 $\alpha=2$를 사용하고, 암호 설계자들은 $\alpha=3$을 사용한다고 하면서, 이 공격의 복잡도를 측정하는데 $\alpha=3$을 적용하였다. 하지만, 암호 설계자이든 암호 해독가이든 항상 최악의 경우를 가정하여 그 경우에도 해당 알고리즘이 안전하다는 것을 증명해야 한다고 생각하며, 따라서 $\alpha=2$를 적용해야 한다고 생각한다. 저자들에게 몇 가지 질문을 던지고 싶은 심정이다. 왜 [TW12a]의 해당 문장을 인용하면서 그 바로 다음 문장에서 항상 최악의 경우에 대한 복잡도를 고려해야 한다는 사실은 간과하였는지, 그리고 저자들은 $\alpha=3$을 사용하였지만, RGB를 공격하려는 사람(e.g. 우리 연구진들)은 $\alpha=2$를 사용하여도 되는 것인지를 말이다. 게다가 [TW12b]에 의하면 q가 짝수일 때, 식이 m개이고 변수가 n개인 underdetermined system($m<n$인 시스템)를 푸는 복잡도는 식과 변수가 모두 $m-\lfloor\frac{n}{m}\rfloor+1$개인 다변수 시스템을 푸는 복잡도와 같다고 한다. 이 결과를 통한 공격을 시도할 경우 복잡도는 조금 더 낮아진다. RGB의 저자들은 이 결과 역시 고려하지 않았다. 이 결과를 고려할 경우 g값을 어떻게 설정하여야 하는지에 대해 뒤에서 자세히 언급하도록 하겠다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

(3) RGB 서명 알고리즘에 대한 키 복구 공격

다변수 다항식 기반 암호 시스템에 대한 키 복구 공격에서는 $P = S \circ F \circ T$에서 양변의 왼쪽과 오른쪽에 S^{-1}, T^{-1}를 각각 취해준 $F = S^{-1} \circ P \circ T^{-1}$라는 관계에서 출발한다. 간단한 표기를 위해 어떤 특정한 함수의 역함수를 $\tilde{}$를 씌워 표기하도록 하겠다. 즉, $S^{-1} = \tilde{S}$, $T^{-1} = \tilde{T}$이다. 이제 $F = \tilde{S} \circ P \circ \tilde{T}$라는 관계에서, 아래의 식을 유추할 수 있다.

$$F^{(k)} = \widetilde{T}^t \left(\sum_{j=1}^{m} \widetilde{s_{kj}} P^{(j)} \right) \widetilde{T}$$

여기서, $\widetilde{s_{kj}}$는 $\tilde{S}$의 k행 j열의 성분이며, $P^{(j)}$는 P의 j번째 성분의 계수행렬, $F^{(k)}$는 F의 k번째 부분함수의 계수행렬을 말한다. 따라서 $F^{(k)}$의 $x_i x_j$의 성분 $f_{ij}^{(k)}$에 대해 다음과 같은 식이 성립한다.

$$f_{ij}^{(k)} = \sum_{x=1}^{m} \sum_{y=1}^{n} \sum_{z=1}^{n} P_{yz}^{(x)} \widetilde{s_{kx}} \widetilde{t_{yi}} \widetilde{t_{zj}}$$

여기서, $P_{xy}^{(z)}$는 $P^{(z)}$의 x행 y열의 성분을 말한다. F는 쉽게 역원을 구할 수 있도록 하기 위해 대부분 특별한 구조를 갖고 있다. 따라서 상당수의 $f_{ij}^{(k)}$ 값이 알려져 있으며, 공개키의 일부인 $P_{xy}^{(z)}$ 역시 알려져 있다. $\tilde{S}, \tilde{T}$의 성분들의 값은 알려져 있지 않으므로 위의 식은 다변수 3차식이라고 할 수 있다. 일반적으로 다변수 다항식 시스템 기반 암호에서의 키 복구 공격은 위에서 얻는 시스템의 해를 구하여 S, T를 복구하는 공격이라고 할 수 있다.

이제 동치 키에 대해 정의하도록 한다.

정의 7. (동치 키) 두 개의 affine 변환 $S: F_q^m \mapsto F_q^m$, $T: F_q^n \mapsto F_q^n$와 2차식 변환 $F: F_q^n \mapsto F_q^m$에 대해 $P = S \circ F \circ T: F_q^n \mapsto F_q^m$를 공개키라고 하자. 이 때 두 개의 affine 변환 $S' : F_q^m \mapsto F_q^m$, $T' : F_q^n \mapsto F_q^n$와 2차식 변환 $F' : F_q^n \mapsto F_q^m$에 대해 $P = S' \circ F' \circ T'$를 만족하며 F'가 F가 갖고 있는 성질을 똑같이 갖고 있어 (S', F', T')를 통해 서명을 위조할 수 있을 경우, (S', F', T')를 (S, F, T)의 **동치 키**라고 한다.

적당한 affine 변환 $\Sigma: F_q^m \mapsto F_q^m$, $\Omega: F_q^n \mapsto F_q^n$에 대해 $(S \circ \Sigma^{-1}, \Sigma \circ F \circ \Omega, \Omega^{-1} \circ T)$는 (S, F, T)의 동치 키이다. $F' = \Sigma \circ F \circ \Omega$가 위의 동치 키의 정의를 만족하도록 하는 Σ, Ω는 굉장히 많으므로 (S, F, T)의 동치 키 역시 굉장히 많다. 여러 다변수 다항식 기반 암호 스킴의 동치키에 대해서는 [WP05]에서 다루고 있다. 공통점은 동치키의 개수가 굉장히 많다는 것이며, 따라서 위에서 얻는 다변수 3차식 시스템의 해의 개수 역시 굉장히 많다. 이 사실을 이용하여 키 복구 공격의 복잡도를 줄일 수 있다[WP05].

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Ding 등은 [DYCC08]에서 [WP05]에서 언급한 동치 키를 빠르게 찾을 수 있는 방법을 언급하였다. 키 아이디어는 수차례의 변환을 행하여 해당 동치 키의 한 열만을 갖는 새로운 다변수 다항식 시스템을 얻을 수 있다는 것이다. Thomae 등은 [TW12b]에서 위의 [DYCC08]의 과정을 "good key를 찾는다"고 표현하였다. good key의 정의는 아래와 같다.

정의 8. (good key). affine 변환들 $S, S' : F_q^m \mapsto F_q^m$, $T, T' : F_q^n \mapsto F_q^n$와 2차식 변환 $F, F' : F_q^n \mapsto F_q^m$에 대해 $P = S \circ F \circ T : F_q^n \mapsto F_q^m$가 공개키이고, (S', F', T')가 (S, F, T)의 동치 키라고 하자. 이 때 두 개의 affine 변환 $S'' : F_q^m \mapsto F_q^m$, $T'' : F_q^n \mapsto F_q^n$와 2차식 변환 $F'' : F_q^n \mapsto F_q^m$에 대해 $P = S'' \circ F'' \circ T''$를 만족하며 F''가 F가 갖고 있는 성질의 일부만을 똑같이 갖고 있으며 S'', T''의 특정 부분이 S', T'와 같을 경우, (S'', F'', T'')를 (S', F', T')에 대응하는 (S, F, T)의 good key라고 한다.

동치 키를 찾을 때 사용되었던 식이 good key를 찾을 때에도 그대로 사용된다. 만약 good key를 찾으면 나머지 식들에 그 값을 대입하면 S', T'의 성분에 관한 상당수의 1차식을 얻을 수 있으며, 따라서 해당하는 값들을 순차적으로 쉽게 구할 수 있게 된다. 이것이 [TW12b]에서 제안된 Thomae-Wolf의 공격이다. 이 보고서의 여러 부분에서의 "키 복구 공격"이라는 용어는 특히 이 Thomae-Wolf의 공격을 지칭하는 것이다.

하지만, RGB의 저자들은 [TW12b]의 한 문장을 인용하여 이 공격에 대한 안전성에 대해 연구하지 않았다. [TW12b]는 Enhanced TTS, Enhanced STS 등의 good key를 3 레이어를 갖는 Rainbow처럼 생각하여 good key를 찾는 공격을 시도하였는데, 거기에서 "이 논문에서의 공격을 UOV같은 레이어가 1개인 스킴에는 적용할 수 없다"라고 서술하였다. 하지만 Thomae는 그의 박사학위논문[Tho13]에서 버젓이 UOV의 good key의 모양을 서술하고 있다. 하지만, 이 good key를 찾는 복잡도는 굉장히 크다. 즉, 위 문장이 뜻하는 것은 UOV에 키 복구 공격을 적용할 수 없는 것이 아니라, 레이어가 여럿인 스킴에 적용했던 키 복구 공격 아이디어를 UOV 같은 레이어가 1개인 스킴의 키 복구 공격의 굉장히 큰 복잡도를 줄이기 위해 적용할 수 없다는 것일 것이다. 하지만 RGB의 저자들은 direct attack에서 언급한 것과 마찬가지로 이 한 문장만을 인용하여 UOV와 같이 레이어가 1개인 RGB 서명 알고리즘에 대한 키 복구 공격은 불가능하다고 서술하였다.

하지만, 우리 연구진은 [SSS10, SSS11] 등의 mixed type의 서명 알고리즘과 그에 대한 공격[BBDM10]에 대해 이미 연구하고, 이 공격에 견디는 다른 mixed type 다변수 다항식 기반 서명 알고리즘을 이미 설계하였었다. 해당 서명 알고리즘은 Oil-Vinegar 구조를 이용한다는 점에서 RGB와 상당히 유사하였으며, 이 서명 알고리즘이 키 복구 공격에 상당히 취약하다는 결과를 얻었다. 이런 정보를 자체적으로 가지고 있는 상황에서 RGB 서명 알고리즘에 대해 접하게 되었다. 우리 연구진은 당연하게도 RGB 서명 알고리즘 역시 키 복구 공격에 취약할 것이라고 생각하여 이를 시도하였다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

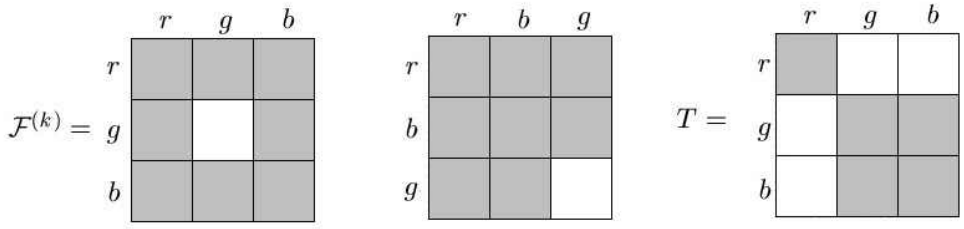


그림 3. RGB 서명 알고리즘의 비밀키의 모양

RGB에 대한 키 복구 공격을 실행하기에 앞서 RGB의 비밀키의 계수행렬의 모양을 그림 3과 같이 표기하였다. 각 행렬의 i행 j열 성분은 u_iu_j(여기서 u_i란 i번째 변수)의 계수를 나타낸다. 여기서는 $y_1, \cdots, y_r, x_1, \cdots, x_{g+b}$의 순서로 변수를 표기하였다. 각 그림의 회색 부분은 임의의 수가 있는 곳이고, 흰색 부분은 0행렬을 말한다. 즉, F의 각 성분의 가운데가 0으로 이루어져 있어 UOV와 다른 모양으로 생각될 수 있지만, 변수의 순서를 바꾸어서 그린 가운데의 그림을 보면 UOV와 같은 모양을 갖고 있다는 것을 알 수 있다. 이 키를 찾기 위한 키 복구 공격을 시도할 경우 식의 개수가 $\frac{g^2(g+1)}{2}$개이고, 변수가 $r^2+g^2+(g+b)^2$개인 다변수 시스템을 얻는다. 이 시스템은 굉장히 복잡하여 direct attack에 비해 복잡도가 크게 높다.

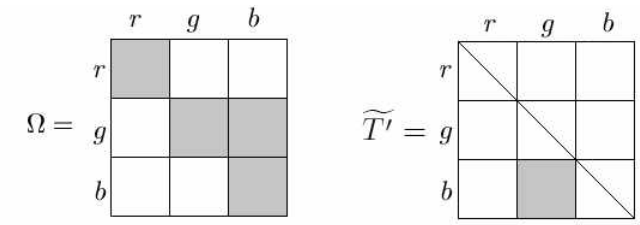


그림 4. RGB 동치 키 $(S', F', T') = (I, F', T')$에 대해 $\widetilde{T'} = \widetilde{T} \circ \Omega$와 Ω의 모양

복잡도를 낮추기 위해 동치 키의 모양을 찾도록 하겠다. 우선 RGB의 경우 F가 1개의 레이어만을 갖기 때문에 임의의 블록별로 설계된 Ω에 대해 $F^{(k)} \circ \Omega$는 모두 같은 모양을 같는다. 따라서 Σ는 임의의 affine 변환이기만 하면 되므로, 최대한 간단한 S'를 선택하기 위해 $\Sigma = S$로 선택하면 $S' = S \circ \Sigma^{-1} = S \circ S^{-1} = I$가 된다. 이 사실은 S가 RGB를 안전하게 하는데 아무런 영향을 주지 않는다는 것을 뜻하며, 따라서 S가 없어져도 RGB의 안전성에는 아무런 문제가 없다. 실제로 RGB와 유사한 UOV는 $P = F \circ T$의 구조로 S의 위치에 어떤 함수도 없다. 한편, $F^{(k)} \circ \Omega$가 모두 $F^{(k)}$와 같은 모양을 갖도록 선택해야 하는데 이 경우 위의 그림처럼 Ω를 선택하면 그림 4와 같은 모양의 $\widetilde{T'}$를 얻는다. 그림 4에서 대각선이 있는 부분에는 1이 있다. 회색 부분은 임의의 수가 있는 것이고 흰색 부분은 0이 있는 것이다. 이를 아래의 보조정리에서 증명하였다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

보조정리 5. RGB 서명 알고리즘에서는, $S'=I_{g\times g}$(g차 정사각행렬)이고 T'가 그림 4의 형태를 만족하는 동치 키를 높은 확률로 찾을 수 있다.

증명. $S'=I_{g\times g}$로 선택할 수 있음을 이미 위에서 설명하였으므로, Ω와 T'를 선택하는 것에 더 집중하도록 하겠다. Ω를 아래와 같이 선택하면 각 $F^{(k)}$와 각 $F^{(k)}\circ\Omega$는 같은 모양을 갖는다.

$$\Omega=\begin{pmatrix}\Omega_{r\times r}^{(1)} & 0 & 0\\ \Omega_{g\times r}^{(2)} & \Omega_{g\times g}^{(3)} & \Omega_{g\times b}^{(4)}\\ \Omega_{b\times r}^{(5)} & 0 & \Omega_{b\times b}^{(6)}\end{pmatrix}$$

이 경우 $\tilde{T}$를

$$\tilde{T}=\begin{pmatrix}\widetilde{T_{r\times r}^{(1)}} & 0 & 0\\ 0 & \widetilde{T_{g\times g}^{(2)}} & \widetilde{T_{g\times b}^{(3)}}\\ 0 & \widetilde{T_{b\times g}^{(4)}} & \widetilde{T_{b\times b}^{(5)}}\end{pmatrix}$$

라고 하면,

$$\tilde{T}\Omega=\begin{pmatrix}\widetilde{T_{r\times r}^{(1)}} & 0 & 0\\ 0 & \widetilde{T_{g\times g}^{(2)}} & \widetilde{T_{g\times b}^{(3)}}\\ 0 & \widetilde{T_{b\times g}^{(4)}} & \widetilde{T_{b\times b}^{(5)}}\end{pmatrix}\begin{pmatrix}\Omega_{r\times r}^{(1)} & 0 & 0\\ \Omega_{g\times r}^{(2)} & \Omega_{g\times g}^{(3)} & \Omega_{g\times b}^{(4)}\\ \Omega_{b\times r}^{(5)} & 0 & \Omega_{b\times b}^{(6)}\end{pmatrix}=\begin{pmatrix}\widetilde{T^{(1)}}\Omega^{(1)} & 0 & 0\\ \widetilde{T^{(2)}}\Omega^{(2)}+\widetilde{T^{(3)}}\Omega^{(5)} & \widetilde{T^{(2)}}\Omega^{(3)} & \widetilde{T^{(2)}}\Omega^{(4)}+\widetilde{T^{(3)}}\Omega^{(6)}\\ \widetilde{T^{(4)}}\Omega^{(2)}+\widetilde{T^{(5)}}\Omega^{(5)} & \widetilde{T^{(4)}}\Omega^{(3)} & \widetilde{T^{(4)}}\Omega^{(4)}+\widetilde{T^{(5)}}\Omega^{(6)}\end{pmatrix}$$

로 계산된다. $\tilde{T}\Omega$가 위 그림의 $\widetilde{T'}$의 모양이 되도록 하기 위한 Ω의 각 블록을 구하기 위해 연립하면 아래와 같은 결과를 얻는다.

$$\Omega^{(1)}=\left(\widetilde{T^{(1)}}\right)^{-1},\ \Omega^{(2)}=0,\ \Omega^{(3)}=\left(\widetilde{T^{(2)}}\right)^{-1},\ \Omega^{(4)}=\left(\widetilde{T^{(2)}}\right)^{-1}\widetilde{T^{(3)}}\Omega^{(6)},$$
$$\Omega^{(5)}=0,\ \Omega^{(6)}=\left(\widetilde{T^{(5)}}-\widetilde{T^{(4)}}\left(\widetilde{T^{(2)}}\right)^{-1}\widetilde{T^{(3)}}\right)^{-1}$$

즉, 위와 같이 Ω를 설정하면 그림과 같은 모양의 동치 키를 얻는다. 역행렬을 요구하는 행렬들이 가역일 확률은 굉장히 높다. 따라서 높은 확률로 그림과 같은 모양의 동치 키가 존재한다. ■

위에서 구한 동치 키를 적용하면, F'가 F와 같은 모양을 갖기 때문에 식의 개수는 그대로 $\frac{g^2(g+1)}{2}$개인 반면, $\tilde{S},\tilde{T}$의 알려지지 않은 부분이 $\tilde{T}$의 아래 가운데 블록의 gb개 뿐이므로 변수의 개수가 gb개로 굉장히 줄어들었다. 저자들이 80비트 안전성을 갖는다고 주장한 파라미터 $(q,r,g,b)=(256,20,24,10)$의 경우 이 동치 키를 찾는 복잡도 역시 80비트이다. 이런 모양의 동치 키를 가질 경우

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$F'^{(k)} = \begin{pmatrix} A^{(1)}_{r\times r} & A^{(2)}_{r\times g} & A^{(3)}_{r\times b} \\ A^{(2)^{\mathrm{T}}} & 0 & A^{(4)}_{g\times b} \\ A^{(3)^{\mathrm{T}}} & A^{(4)^{\mathrm{T}}} & A^{(5)}_{b\times b} \end{pmatrix}$$

라고 한다면, $A^{(1)}, A^{(3)}, A^{(5)}$ 부분은 공개키 $P^{(k)}$의 해당 부분과 같은 값을 갖게 된다. 즉, RGB 스킴의 공개키는 공격자에게 비밀키의 상당 부분을 알려주고 있는 셈이다. 이런 동치 키를 더 빠르게 찾기 위해서 good key를 찾아보도록 한다.

우선 $S' = I$이므로 역시 $S'' = I$로 설정하면 충분하다. 이제 T''의 모양을 찾기 위해 그림 5와 같은 모양의 Ω'를 사용한다.

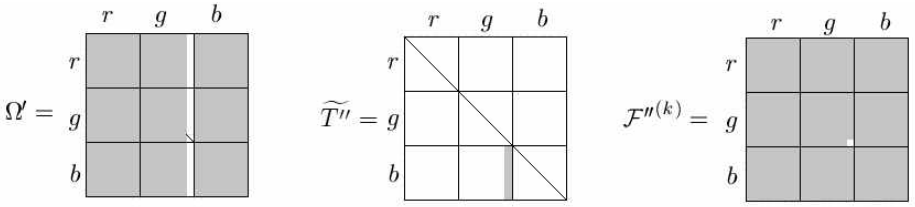


그림 5. RGB good key $(S'', F'', T'') = (I, F'', T'')$에 대해 $\widetilde{T}'' = \widetilde{T}' \circ \Omega'$와 Ω', 각 $F''^{(k)}$의 모양

Ω'는 $\widetilde{T}'' = \widetilde{T}' \circ \Omega'$의 일부분이 $\widetilde{T}'$와 같으면서도, 최대한 간단하게 되어야 한다. 그러기 위해서 Ω'의 각 열들은 아래의 두 가지 유형 중 하나가 된다.

(i) 모든 항이 임의의 항이다. 이는 $\widetilde{T}''$의 해당 열을 최대한 간단한 형태로 만든다.

(ii) 대각 성분에 있는 항만 1이고, 나머지 항은 0이다. 이는 $\widetilde{T}''$의 해당 열이 $\widetilde{T}'$의 해당 열과 같아지게 한다.

$\widetilde{T}''$를 최대한 간단한 형태가 되도록 하기 위해서는 Ω'에 (ii)에 해당하는 열이 1개만 있어야 한다. 그 이유는 Ω'의 모든 열이 (i)의 모양일 경우 $\widetilde{T}'' = I$가 되어 $P = F''$가 되어 (S'', F'', T'')가 good key의 정의에 어긋나게 된다. 따라서 (ii)에 해당하는 열이 최소 1개는 있어야 한다. 그렇다면 어떤 열이 (ii)에 해당하는 열이어야 할까? j_1번째 열이 (ii)에 해당하는 열이라고 하자. 이 경우 $\widetilde{T}'$의 j_1번째 열과 $\widetilde{T}''$의 j_1번째 열이 같아야 하므로, good key의 정의를 만족하기 위해서는 $r+1 \le j_1 \le r+g$이어야 한다. 우리는 $j_1 = r+g$로 선택하였다. 이제 해당 good key가 존재한다는 사실을 증명하도록 하자.

보조정리 6. (S', F', T')를 위 보조정리에서 얻은 RGB 스킴의 동치 키라고 하자. 그러면 $S'' = I_{g\times g}$, 그리고 T''가 위 그림과 같은 모양을 갖는 RGB 스킴의 동치 키에 대응하는 good key (S'', F'', T'')가 존재한다.

증명. $S' = I_{g\times g}$이므로 $S'' = I_{g\times g}$로 정하는 것은 당연하다. 이제 Ω', 그리고 $\widetilde{T}''$의 설정에 주목하도록 하자. Ω'가 왜 위 그림의 모양처럼 설계되었는지는 위에서 이미 설명하였다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

이제 이런 Ω'에 대해 $\widetilde{T''}=\widetilde{T'}\circ\Omega'$가 위 그림의 모양을 갖도록 설계할 수 있는지를 증명하도록 하겠다. 우선 $\widetilde{T'}$, Ω'를 아래와 같이 쓰도록 하자.

$$\widetilde{T'}=\begin{pmatrix} I_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times 1} & 0_{(r+g-1)\times b} \\ 0_{1\times(r+g-1)} & 1 & 0_{1\times b} \\ \widetilde{T'^{(1)}_{b\times(r+g-1)}} & \widetilde{T'^{(2)}_{b\times 1}} & I_{b\times b} \end{pmatrix}, \Omega'=\begin{pmatrix} \Omega'^{(1)}_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times 1} & \Omega'^{(2)}_{(r+g-1)\times b} \\ \Omega'^{(3)}_{1\times(r+g-1)} & 1 & \Omega'^{(4)}_{1\times b} \\ \Omega'^{(5)}_{b\times(r+g-1)} & 0_{b\times 1} & \Omega'^{(6)}_{b\times b} \end{pmatrix}$$

이 경우, 아래와 같이 계산할 수 있다.

$$\widetilde{T'}\Omega'=\begin{pmatrix} I_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times 1} & 0_{(r+g-1)\times b} \\ 0_{1\times(r+g-1)} & 1 & 0_{1\times b} \\ \widetilde{T'^{(1)}_{b\times(r+g-1)}} & \widetilde{T'^{(2)}_{b\times 1}} & I_{b\times b} \end{pmatrix}\begin{pmatrix} \Omega'^{(1)}_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times 1} & \Omega'^{(2)}_{(r+g-1)\times b} \\ \Omega'^{(3)}_{1\times(r+g-1)} & 1 & \Omega'^{(4)}_{1\times b} \\ \Omega'^{(5)}_{b\times(r+g-1)} & 0_{b\times 1} & \Omega'^{(6)}_{b\times b} \end{pmatrix}$$
$$=\begin{pmatrix} \Omega'^{(1)} & 0 & \Omega'^{(2)} \\ \Omega'^{(3)} & 1 & \Omega'^{(4)} \\ \widetilde{T'^{(1)}}\Omega'^{(1)}+\widetilde{T'^{(2)}}\Omega'^{(3)}+\Omega'^{(5)} & \widetilde{T'^{(2)}} & \widetilde{T'^{(1)}}\Omega'^{(2)}+\widetilde{T'^{(2)}}\Omega'^{(4)}+\Omega'^{(6)} \end{pmatrix}$$

이제 $\widetilde{T''}$가 그림과 같은 모양을 갖게 하려면, Ω'의 $r+g$번째 열을 제외한 부분행렬이

$$\begin{pmatrix} \Omega'^{(1)} & \Omega'^{(2)} \\ \Omega'^{(3)} & \Omega'^{(4)} \\ \Omega'^{(5)} & \Omega'^{(6)} \end{pmatrix} = \left(\widetilde{T'}\right)^{-1}\begin{pmatrix} I_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times b} \\ 0_{1\times(r+g-1)} & 0_{1\times b} \\ \widetilde{T'^{(1)}_{b\times(r+g-1)}} & I_{b\times b} \end{pmatrix}=\begin{pmatrix} I_{(r+g-1)\times(r+g-1)} & 0_{(r+g-1)\times b} \\ 0_{1\times(r+g-1)} & 0_{1\times b} \\ -\widetilde{T'^{(1)}_{b\times(r+g-1)}} & I_{b\times b} \end{pmatrix}$$

와 같이 설정되면 된다. 한편 Ω'를 이와 같이 설정할 경우 $F''^{(k)}=F'^{(k)}\circ\Omega'$의 $(r+g)\times(r+g)$성분은 $F'^{(k)}$의 해당 성분과 같은 값을 갖게 되므로, 0이 되어 good key의 조건을 만족하게 된다. ■

위의 보조정리 6에서의 good key를 찾기 위해서는 식의 개수가 F''의 성분의 개수와 같은 g개이고, 변수의 개수는 b개인 다변수 2차식 시스템을 풀어야 한다. 해당 시스템의 복잡도는 특히 b에 영향을 많이 받아 b가 작을수록 그 복잡도가 현저히 낮아지며, 또한 r과는 상관이 없다. 그런데 [TLS16]을 보면 b가 작을수록 RGB의 속도가 빨라지기 때문에, b를 상대적으로 작게 설정하였다. 이 경우 키 복구 공격에는 상당히 낮은 복잡도를 갖게 되며, 해당 복잡도는 아래의 표 12에 기록해두었다. 아래의 표 12에서 $q=256=2^8$로 선택하였다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

파라미터 (r,g,b)	저자들이 주장한 복잡도 [ST15, TLS16]	키 복구 공격 복잡도$(\alpha=2/\alpha=2.8)$	실제 공격 시간(초)
(20, 24, 10)	2^{80}	$2^{20}/2^{26}$	0.54
(16, 18, 10)	2^{64}	$2^{23}/2^{30}$	0.48
(24, 24, 12)	2^{80}	$2^{25}/2^{33}$	1.7
(30, 32, 16)	2^{96}	$2^{32}/2^{43}$	90.68
(32, 34, 18)	2^{118}	$2^{34}/2^{46}$	660.37
(40, 35, 20)	2^{128}	$2^{39}/2^{53}$	24551.16

표 12. RGB에 대한 키 복구 공격의 이론적 복잡도와 실제 공격시간

표를 보면 저자들이 주장한 복잡도에 비해 이론적인 복잡도는 훨씬 낮으며, 실제로 good key를 찾는데 많은 시간이 걸리지 않았다는 사실을 알 수 있다. 동치 키를 찾는 시스템의 각 다항식은 $S'=I$이기 때문에 아래와 같은 2차식이다.

$$f'^{(k)}_{ij}=0=\sum_{x=r+g+1}^{n}\sum_{y=r+g+1}^{n}P^{(k)}_{xy}\widetilde{t_{xi}}\widetilde{t_{yj}}+\sum_{y=r+g+1}^{n}P^{(k)}_{iy}\widetilde{t_{yj}}+\sum_{x=r+g+1}^{n}P^{(k)}_{xj}\widetilde{t_{xj}}$$

good key를 찾는다는 것은 $\widetilde{t_{x,r+g}}$ 형태의 변수를 모두 찾는 것을 의미한다. 따라서 위 식의 $j=r+g$인 식들에 대입할 경우 모두 1차식이 된다. 1차식이 되는 식의 수는 $g(g-1)$개이며, 남은 변수의 수는 $b(g-1)$개다. 따라서 $g>b$일 경우에는 식의 수가 변수의 수보다 많으므로 Gauss 소거법을 이용해 다항식 시간 안에 나머지 변수들을 모두 찾을 수 있으므로 good key를 찾는 복잡도가 사실상 동치 키를 찾는 복잡도라고 할 수 있다.

지금부터 앞에서 설명한 RGB의 공격방법을 알고리즘으로 정리하고 이 알고리즘을 구현하여 테스트한 결과를 정리할 것이다. RGB의 공격은 공개키 $P=F\circ T$가 주어져 있을 때, F와 T를 찾는 것이다. 그런데 T를 찾으면 $F=P\circ T^{-1}$로부터 F도 찾을 수 있기 때문에 결국 비밀키 T를 찾는 것으로 귀결될 수 있다. 원래 RGB의 T는 $(r+g+b)\times(r+g+b)$의 행렬이지만 역시 앞 절의 결과에 의해서 동치인 $\widetilde{T'}$을 찾으면 되고 이것은 단지 $b\times g$ 부분행렬만 결정하면 되기 때문에 공격 계산량을 많이 줄일 수 있다. 더욱이 앞 절의 Good key 공격을 적용하여 찾아야 할 $b\times g$ 부분행렬의 한 열씩을 순차적으로 찾으면 되기 때문에 공격 계산량을 더욱 줄일 수 있다. $b\times g$ 부분행렬의 한 열씩을 복구하는 것은 순차적으로 하여도 되지만 서로 독립적으로 찾아도 된다. 그러나 이미 한 열씩을 복구하는 계산량이 매우 적기 때문에 순차적으로 한 열씩 복구하는 방법을 택하였다. $\widetilde{T'}$의 $b\times g$ 부분행렬의 각 열을 복구하는 것은 b개의 변수를 가지는 g개의 2차 연립방정식을 푸는 문제로 귀결되어서 이것을 gröbner 기저방법을 이용하여 푼 후, 각 열을 복구한다. $b\times g$ 부분행렬의 모든 열을 복구해야 하기 때문에 이러한 과정을 g번 반복한다. 이러한 내용을 정리한 알고리즘은 다음과 같다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

RGB 키 복구 공격
입력: RGB의 공개키 $P(=F \circ T)$
출력: RGB의 동치 비밀키 $\widetilde{T'}$
1: $i=1$부터 g까지 아래의 스텝2번에서 8번까지를 반복한다.
2: b개의 변수 $x_1,...,x_b$를 가지는 다변수 다항식환 P을 정의한다.
3: 다항식환 P에서 정의된 $n \times n$ 항등행렬 I을 정의한다.
4: $x_1,...,x_b$를 원소로 가지는 $b \times 1$ 행렬 T_i을 정의한다.
5: T_i를 항등행렬 I의 $r+g+1$행 $r+i$열에 넣은 행렬을 I_i라 정의한다.
6: I_i의 1행 $r+i$열 위치에서 시작되는 $n \times 1$ 행렬 U_i을 정의한다.
7: $U_i^T \cdot P \cdot U_i$로부터 나온 식으로 정의된 다항식환 P에서의 아이디얼 M_i을 정의한다.
8: M_i의 Gröbner 기저를 계산하여 M_i의 variety V_i를 계산한다.
9: V_i들로부터 $\widetilde{T'}$를 복원한다.

표 13. RGB 키 복구 공격

이 알고리즘은 Magma [BCP97]를 이용하여 Intel Xeon E5-2687W CPU 3.1 GHz, 256GB RAM 메모리를 가지는 하드웨어에서 구현하여 테스트를 수행하였다. 그 결과는 위의 표 12와 같다.

(4) 안전한 RGB 서명 알고리즘의 파라미터

앞에서 우리는 RGB 서명 알고리즘이 저자들이 생각하는 것보다 direct attack이나 키 복구 공격에 취약하다는 사실을 살펴보았다. 하지만 이들은 다항식 시간 공격이 아니기 때문에 파라미터를 충분히 늘리면 공격복잡도가 늘어나 요구하는 안전성을 만족하게 된다.

(가) Direct Attack

이 공격에 대한 안전성을 살펴보기에 앞서 먼저 앞에서 언급한 [TW12a]의 결과를 살펴볼 필요가 있다.

정리 4[TW12a]. 우리는 $q=2^k$개의 원소를 가진 유한체 상에서 정의된 m개의 식과 n개의 변수를 가진 다변수 2차식 시스템에 대해 생각하고 있다고 하자($m<n$). 많은 실험 결

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

과, 이 다변수 2차식 시스템의 해를 찾는 것은 같은 유한체에서 정의된 $m-\lfloor \frac{n}{m} \rfloor +1$개의 식과 변수를 갖는 다변수 2차식의 해를 찾는 것과 실질적으로 비슷한 복잡도를 갖는다.

위의 결과에 의하면 이 공격에 대한 복잡도는 사실상 공개키의 식의 개수인 g에 의존한다고 말할 수 있다. 우리는 $\alpha=2$일 경우에 Hybrid F_5알고리즘을 실행하였을 때 해당 복잡도를 갖는 determined system(식과 변수의 개수가 같은 시스템)의 식의 개수(=변수의 개수)의 하한을 조사하여 아래 표에 적어두었다.

λ	80	112	118	128	192	256
식의 개수	27	39	41	45	70	96

표 14. $q=256$일 때 해당 복잡도를 달성하기 위해 필요한 Determined System의 식의 개수

(나) 키 복구 공격

우리는 앞에서 키 복구 공격이 g개의 식과 b개의 변수를 갖는 다변수 2차식 시스템의 해를 찾는 복잡도와 같다는 사실을 보였다. 따라서 키 복구 공격에 대해 안전하게 하기 위해 g와 b를 어떤 조건이 되도록 설정할지에 대해 고려해보았다.

우선, 위의 표를 참조해서 위의 Direct Attack에 어느 정도 안전할 정도로 g를 설정했다고 가정하자. 이 경우 $b<g$로 b를 선택하면, 키 복구 공격의 복잡도가 위의 표에서 얻은 복잡도보다 낮아지게 된다. 따라서 $b \geq g$가 되도록 b를 선택하는 것이 바람직하다.

(다) UOV 공격

앞에서 잠깐 언급한대로 $g<r+b<\frac{g(g+1)}{2}$를 만족한다면 이 공격에 대한 RGB 서명 알고리즘의 복잡도는 $q^{r+b-g-1}\cdot g^4$이다.

(라) 위의 공격들에 안전한 파라미터 선택

위에서 언급한 공격들에 안전한 파라미터 선택을 하기 위해서는 다음과 같은 과정을 통해 선택한다.

- (가)를 적용하여 주어진 λ에 대한 안전성을 갖는 determined system의 식의 개수로 g를 선택한다.
- (나)를 참고하여 키 복구 공격에 안전하도록 $b(\geq g)$를 선택한다.
- (다)를 참고하여 UOV 공격에 안전하도록 r을 선택한다.
- 선택된 (r,g,b)에 대해, 이 파라미터가 direct attack에 안전하지 않다면 g를 1 키운

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

뒤 두 번째 과정으로 돌아간다.

이 과정을 통하여 선택된 파라미터 및 선택된 RGB의 각 공격들에 대한 안전성을 아래의 표에 담았다.

λ	추천된 파라미터 (r,g,b)	키 복구 공격	UOV 공격	direct attack
64	(16, 24, 31)	2^{130}	2^{66}	2^{69}
80	(20, 29, 38)	2^{161}	2^{83}	2^{80}
96	(24, 35, 46)	2^{191}	2^{100}	2^{96}
118	(30, 43, 57)	2^{237}	2^{125}	2^{119}
128	(32, 47, 62)	2^{253}	2^{134}	2^{130}

표 15. RGB에 대해 주어진 보안레벨을 만족하는 추천 파라미터($q=2^8$)

하지만, 위에 주어진 파라미터를 가진 RGB는 더 이상 기존 다변수 다항식 기반 서명 알고리즘에 비해 효과적이지 못하다. 아래는 $\lambda=80,128$에 대해 같은 안전성을 갖는 UOV, Rainbow 서명 알고리즘의 키 길이와 효율성(서명 생성 및 서명 확인시 요구되는 이론적인 곱셈 수)를 비교한 것이다. 아래의 표를 확인해보면, 추천된 파라미터의 RGB는 같은 안전성을 갖는 UOV, Rainbow에 비해 효과적이지 못하다. 따라서, 적어도 RGB는 경량기기에서 사용하기에는 무리인 모델이라고 여겨진다. 즉, RGB는 완전히 깨졌다고 말할 수 있다.

λ	스킴	공개키(KB)	비밀키(KB)	서명 생성	서명 확인
80	UOV(F_{2^8}, 28, 37)	60.457	53.447	62748	63700
	Rainbow(F_{2^8}, 18, 13, 13)	26.279	20.126	22569	27874
	RGB(F_{2^8}, 20, 29, 38)	104.672	101.643	87732	86184
128	UOV(F_{2^8}, 44, 59)	234.609	202.531	237395	244728
	Rainbow(F_{2^8}, 36, 21, 22)	136.055	102.503	113383	142437
	RGB(F_{2^8}, 32, 47, 62)	449.713	442.496	469204	466900

표 16. 다양한 안전도에서 UOV, Rainbow, RGB 서명 알고리즘의 키 길이, 효율성 비교

2. Rainbow 안전성 분석

가. UOV 공격

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Unbalanced Oil-and-Vinegar(UOV) 공격 [KS98]은 다변수 이차식 기반 암호시스템에 사용되는 변수를 O와 V의 두 집합으로 나눌 수 있고, O에 속하는 변수들로 이루어진 이차항이 나타나지 않을 때 이루어질 수 있다. Rainbow 전자서명은 일종의 다계층 UOV 전자서명이므로 마지막 계층의 O에 속하는 변수를 oil 변수, V에 속하는 변수를 vinegar 변수라고 하자. 또한 |V|=v, |O|=o, n=v+o라고 정의하면 UOV 공격은 다음과 같다.

Rainbow 전자서명에 대한 UOV 공격

입력: 행렬 $P_i \in F_q^{m \times n}$ for $i = 1,...,m, O = \{x = (x_1,...,x_n) \in F_q^n \mid x_1 = ... = x_v = 0\}$

출력: T와 $F' = S \circ F$

1: m개의 랜덤한 원소 $\lambda_i \in F_q$을 선택한 후, λ_i들에 대하여 행렬 $\sum_{i=1}^{m} \lambda_i P_i$를 W이라고 놓고 가역행렬인 P_j에 대하여 $\widehat{W} = P_j^{-1} \cdot W$를 계산한다.

2: $\widehat{W}$의 invariant 부분공간을 구한 후, 각 부분공간이 $M_T^{-1} \cdot O$의 부분공간인지를 확인한다.

3: $M_T^{-1} \cdot O$의 기저를 찾을 때 까지 위의 과정을 반복하여 T를 구한다.

4: $P \circ T^{-1}$로부터 F'을 계산한다.

표 17. Rainbow 전자서명의 UOV 공격 알고리즘

UOV 공격의 핵심 아이디어는 비밀함수의 모든 컴포넌트 함수에 O에 속하는 변수들로 이루어진 이차항이 나타나지 않기 때문에 임의의 두 개의 컴포넌트 함수 F_i, F_j에 대하여 $F_i^{-1}(F_j(O))$가 일정확률로 다시 O의 부분공간이 된다는 것을 이용하는 것이다. 물론 비밀함수 F_i, F_j는 모르기 때문에 대신에 공개키의 컴포넌트 함수를 이용하여 불변공간을 찾으면 이것이 $M_T^{-1} \cdot O$의 부분공간이 되기 때문에 이로부터 M_T에 해당하는 비밀 아핀변환 T를 찾아낼 수 있다는 것이다. $F_i^{-1}(F_j(O))$가 다시 O의 부분공간이 될 확률은 다음과 같다.

보조정리 7. 역행렬이 존재하는 F_i에 대하여 $F_i^{-1}(F_j(O))$가 O의 부분공간이 될 확률은 $1/q^{(2v-n-1)}$보다 크다.

(증명) $F_i^{-1}F_j$를 ϕ라 하자. 그러면 다음을 알 수 있다.

$$\dim(O) = \dim(\phi(O)) = o$$
$$\dim(V) = \dim(\phi^{-1}(V)) = v$$

이 사실과 $\phi(O) \subseteq \phi^{-1}(V)$로부터 0이 아닌 O의 원소 α에 대해서 다음의 확률을 계산할 수 있다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$\Pr ob(\phi(\alpha)=c\cdot\alpha)=\frac{q-1}{q^v-1}$$

$\phi(\alpha)=c\cdot\alpha$이면 임의의 상수 d에 대해 $\phi(d\cdot\alpha)=c\cdot(d\cdot\alpha)$가 성립하기 때문에 다음의 확률을 얻을 수 있다.

$$\begin{aligned}\Pr ob(\phi(O)\subseteq O) &\geq \Pr ob(\phi(\alpha)=c\cdot\alpha|\ \alpha\in O)\\ &=\sum_{\alpha\in O}\Pr ob(\phi(\alpha)=c\cdot\alpha)\\ &=\frac{(q^o-1)(q-1)}{(q-1)(q^v-1)}\approx q^{o-v-1}\end{aligned}$$

위의 알고리즘의 스텝 3에서 $M_T^{-1}\cdot O$의 부분공간인지의 확인은 다음 사실을 이용하여 이루어질 수 있다.

보조정리 8. 만약 $H\subset M_T^{-1}\cdot O$이면 H의 모든 원소 x,y에 대해서 다음이 성립한다.

$$x^t\cdot P_i\cdot y=0$$

Rainbow 전자서명은 일종의 다계층 UOV 전자서명이므로 Rainbow의 마지막 계층은 위의 UOV 공격이 그대로 적용될 수 있다. 물론 Rainbow의 공개키는 UOV와 다르게 여분의 아핀변환 S가 있어서, $P=S\circ F\circ T$와 같이 계산되는데, $F'=S\circ F$로 본다고 하면 $P=F'\circ T$로서 UOV와 같은 형태를 취하게 된다. 따라서 Rainbow 전자서명의 마지막 계층에서의 vinegar 변수를 v, oil 변수를 o라 한다고 하면 위의 보조정리에 의해서 UOV 공격의 복잡도는 다음과 같다.

$$C_{\text{UOV ※ack Rainbow}}(q,v,o,n)=q^{v-o-1}\cdot o^4$$

나. MinRank 공격

유한체 F_q에서 정의된 비밀함수 $F=(f^{(1)}(x_1,...,x_n),\cdots,f^{(m)}(x_1,...,x_n)):F_q^n\rightarrow F_q^m$이 있을 때, 다변수 이차다항식 $f^{(i)}(x_1,...,x_n)$는 대칭행렬 M_i에 대해서 $(x_1,...,x_n)\cdot M_i\cdot(x_1,...,x_n)^t$로 표현될 수 있다. 그러면 TTS와 Rainbow 암호시스템들은 앞부분의 컴포넌트 함수 $f^{(i)}$들이 뒷부분의 컴포넌트 함수들보다 작은 rank를 가지게 되는데, MinRank 공격은 MinRank를 가지는 비밀함수의 컴포넌트 함수를 찾고 이를 이용하여 비밀함수를 숨기는 아핀변환 S와 T, 최종적으로 비밀함수 F를 찾는 공격이다.

MinRank 공격은 Goubin과 Courtois가 TPM 암호시스템을 공격하기 위하여 처음 제안한 것으로 MinRank 알고리즘이라 불린다. 이것은 다음과 같다 [CSV94].

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Rainbow 전자서명에 대한 MinRank 알고리즘	
입력:	자연수 r, Rainbow 전자서명의 공개키 행렬 M_i $(i=1,...,m)$
출력:	Rainbow 전자서명의 첫 번째 비밀키 행렬 $Q_1,...,Q_{o_1}$
1:	$r=2v_1$이라 놓는다. (여기서 v_1은 첫 번째 계층의 vinegar 변수의 개수)
2:	k개의 랜덤한 벡터 $X_i \in F_q^n$을 선택한다. (여기서 $k=\left\lceil \frac{m}{n} \right\rceil$로 선택한다.) 변수 λ_i들에 대하여 행렬 $\sum_{i=1}^{m}\lambda_i M_i$를 M이라고 놓는다.
3:	$i=1,...,k$에 대하여 $M \cdot X_i = 0$를 만족하는 λ_i를 구한다.
4:	이 과정을 o_1번 반복하여 찾은 행렬을 $Q_1,...,Q_{o_1}$라 놓는다.

표 18. Rainbow 전자서명의 MinRank 알고리즘

위의 알고리즘에서 미지수 λ_i가 n개 있으며, 과정 3에서 $i=1,...,k$에 대하여 $M \cdot X_i = 0$에 의해서 식이 kn개가 생기므로 결국 위의 알고리즘은 MinRank 문제를 n개의 미지수와 kn개의 식으로 이루어진 일차 연립방정식을 푸는 문제로 바꾸게 된다. 이러한 연립방정식은 $k=m/n$으로 택하였기 때문에 만약 해가 존재하게 된다면 그것은 유일하게 되고 따라서 비밀함수의 가장 작은 rank를 가지는 첫 번째 컴포넌트 함수를 찾게 된다. 이러한 연립방정식은 Gauss 소거법을 이용하여 풀면 약 $m^2(nk/2-m/6)+mn^2k)$번의 유한체 곱셈으로 풀 수 있다. 그러나 MinRank 알고리즘의 가장 큰 어려움은 과정 1에서 랜덤한 벡터 $X_i \in F_q^n$를 선택하는 과정이다. 왜냐하면 $M \cdot X_i = 0$를 만족하기 위해서는 랜덤하게 선택한 벡터 X_i가 $\ker(M)$에 대입되어야 하기 때문이다. 알고리즘에서 찾기 원하는 행렬이 $rank(M) < r$을 만족하는 행렬이므로 이런 행렬 M에 대하여 $\ker(M)$은 $n-r$의 차원을 가지게 된다. 따라서 F_q^n의 원소를 랜덤하게 선택했을 때, 그 원소가 $\ker(M)$에 들어갈 확률은 $q^{n-r}/q^n = 1/q^r$이 된다. 이러한 벡터를 k개를 선택해야 하므로 이 벡터들이 모두 $\ker(M)$에 들어갈 확률은 $1/q^{kr}$이 된다. 따라서 MinRank 알고리즘의 복잡도는 $O(q^{kr} \cdot (m^2(nk/2-m/6)+mn^2k))$이 된다.

정리 5. 비밀키의 형태와 상관없이 공개키의 각 컴포넌트함수의 행렬표현들이 최소 랭크 r을 갖는 행렬들의 일차결합 $M=\sum_{i=1}^{m}\lambda_i M_i$이 존재한다면 위의 알고리즘은 $O(q^{kr} \cdot m^2(nk/2-m/6)+mn^2k))$정도의 유한체 곱셈을 통해서 $\ker M$에 들어가는 벡터를 찾을 수 있다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

위의 정리를 Rainbow 전자서명에 적용하여 보면, Rainbow에서는 $k=1$을 가지게 되고, 제일 작은 랭크는 첫 번째 계층의 랭크이므로 $r=2v$를 가지게 된다. 따라서 MinRank 공격의 복잡도는 $O(q^{2v}\cdot m^3)$가 됨을 알 수 있다. 그러나 Rainbow 전자서명에 사용되는 행렬같이 가역행렬 M_T와 $\begin{pmatrix}***0\end{pmatrix}$의 형태를 가지는 $n\times n$행렬 F_i에 대해서 행렬 $M_i=M_T^t\cdot F_i\cdot M_T$들에 대한 MinRank 문제는 다음과 같은 관찰에 의해 더 작은 계산복잡도로 풀려질 수 있다 [BG06].

정리 6. v개의 $\begin{pmatrix}***0\end{pmatrix}$의 형태를 가지는 $n\times n$행렬 F_i(여기서 *는 $v\times v$ 행렬)와 $(0*)$형태의 F_q^n-벡터 w(여기서 0는 F_q^v의 벡터)에 대해서 w가 $M=\sum_{i=1}^{m}\lambda_i M_i$의 kernel에 들어가게 하는 최소 한개의 0이 아닌 스칼라 λ_i가 존재할 확률은 $1/q$보다 크다.

(증명) F_i와 w의 형태 때문에 $F_i\cdot w$는 많아야 v개의 0이 아닌 엔트리를 가진다. w가 유니폼한 랜덤이므로 $F_i\cdot w$들도 유니폼한 랜덤이라고 할 수 있고, 따라서 이러한 벡터 $F_i\cdot w$들이 일차독립일 확률은 다음과 같다.

$$\prod_{i=0}^{v-1}(1-\frac{q^i}{q^v})<(1-\frac{1}{q})$$

따라서 w가 F_i의 비자명한 일차결합의 kernel에 들어갈 확률은 $1/q$보다 크다.

따름정리 1. 가역행렬 M_T와 $\begin{pmatrix}***0\end{pmatrix}$의 형태를 가지는 $n\times n$행렬 F_i에 대해서 행렬 $M_i=M_T^t\cdot F_i\cdot M_T$들과 $2v$에 대한 $2v-$MinRank 문제는 $O(q^{v+1}\cdot m^3)$의 복잡도를 가지고 풀릴 수 있다.

(증명) 벡터 w를 F_q^n에서 임의로 선택하였을 때, $M_T\cdot w$가 $(0*)$형태를 가질 확률은 $1/q^v$이다. 그러면 위의 따름정리에 의해서 w가 $M=\sum_{i=1}^{m}\lambda_i M_i$의 kernel에 들어갈 확률은 $1/q^{v+1}$이 된다. 다음은 MinRank 알고리즘을 따르면 되므로 이러한 MinRank 문제는 $O(q^{v+1}\cdot m^3)$의 복잡도를 가지고 풀릴 수 있다.

다. HighRank 공격

Coppersmith 등에 의해서 제안된 HighRank 공격 [CSV94]은 MinRank 공격의 반대공

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

격이라고 볼 수 있다. MinRank 공격은 작은 rank를 가지는 행렬들을 찾는 공격으로 이러한 행렬들은 반대로 큰 차원의 kernel을 가지게 된다. 따라서 MinRank 공격은 작은 수의 layer에 의해서 공유된 큰 kernel을 찾는 공격이라고 할 수 있다. 큰 kernel이란 0의 값을 가지는 변수들이 적은 변수들의 집합을 의미한다. 따라서 작은 수의 layer에 의해서 공유된 큰 kernel을 찾는다는 것은 적은 수의 변수가 사용된 컴포넌트 함수를 찾는다는 것을 의미한다. 반면에 HighRank 공격은 큰 rank를 가지는 행렬들을 찾는 공격이다. MQ-PKC의 공개키의 컴포넌트 함수의 이차항들의 행렬은 아핀 변환에 의해서 서로 섞이게 되므로 대부분은 full rank를 가지게 된다. 따라서 이러한 행렬들은 일차결합을 하면 많은 경우에 high rank를 가지게 된다. 행렬이 큰 rank를 가지게 되면 작은 차원의 kernel을 가지게 되므로 결국 HighRank 공격은 MinRank 공격과는 반대로 많은 수의 layer에 의해서 공유된 작은 kernel을 찾는 공격이라고 할 수 있다. 즉, 적은 수의 layer에서 사용된 변수를 찾는 공격이라고 할 수 있다. HighRank 공격의 아이디어는 비밀키를 이루는 이차다변수 다항식에서 일부의 변수들은 마지막 layer에 있는 다항식에만 사용된다는 점이다. 즉, 마지막 전의 layer들에서 이러한 변수들이 사용되지 않기 때문에 이러한 layer에서의 다변수 이차 다항식의 행렬들은 full rank를 가지지 않게 된다. 따라서 만약에 이렇게 마지막 layer에서만 사용되는 변수들이 있다면 HighRank 공격을 통해서 마지막 전의 layer를 분리해 낼 수 있는 것이다. m개의 $n \times n$ 행렬F_i은 다음을 만족한다고 하자.

- $rank(F_i) \leq rank(F_{i+1})$
- $rank(F_{m-o}) < rank(F_{m-o+1}) = ... = rank(F_m) = n$

또한 가역행렬 M_T에 대해서 행렬 P_i가 다음과 같이 정의되었다고 하자.

$$P_i = \sum_{j=1}^{m} s_{ij}(M_T^t \cdot F_i \cdot M_T)$$

그러면 다음의 HighRank 알고리즘은 $M_T^t \cdot F_{m-1} \cdot M_T$와 같은 rank를 갖는 P_i들의 일차결합을 찾는 알고리즘이다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Rainbow 전자서명에 대한 HighRank 알고리즘	
입력:	Rainbow의 공개키 행렬 $P_i \in F_q^{m \times n}$ for $i = 1,...,m$
출력:	T^{-1}
1:	m개의 랜덤한 원소 $\lambda_i \in F_q$을 선택한다.
2:	λ_i들에 대하여 행렬 $\sum_{i=1}^{m} \lambda_i P_i$를 P이라고 놓고 $V = \ker(P)$를 계산한다.
3:	만약 $\dim(V) \geq 1$이면 $(\sum_{i=1}^{m} x_i P_i)V = 0$을 만족하는 해 x_i를 구한다.
4:	만약 해집합의 차원이 $m-o$면 그 때의 V의 기저와 O_u를 이용하여 T^{-1}를 계산한다.

표 19. Rainbow 전자서명의 HighRank 알고리즘

위의 알고리즘에 대한 설명은 다음과 같다. 먼저 V를 다시 계산해 보면,

$$\begin{aligned} P &= \sum_{i=1}^{m} \lambda_i P_i = \sum_{i=1}^{m} \lambda_i \sum_{j=1}^{m} s_{ij} M_T^T \cdot F_j \cdot M_T \\ &= M_T^T \cdot (\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{m} s_{ij} F_j) \cdot M_T \\ &= M_T^T \cdot (\sum_{j=1}^{m} \alpha_j F_j) \cdot M_T \quad where \quad \alpha_j = \sum_{i=1}^{m} \lambda_i s_{ij} \end{aligned}$$

여기서 M_T는 가역행렬이므로 $rank(P) = rank(\sum_{j=1}^{m} \alpha_j F_j)$와 같다. 따라서 $\dim(V) = n - rank(\sum_{j=1}^{m} \alpha_j F_j)$가 된다. 그런데 $rank(F_{m-o+1}) = ... = rank(F_m) = nrank(F_m) = n$이므로 $\dim(V) \geq 1$을 만족하기 위해서는 $\alpha_{m-o+1} = ... = \alpha_m = 0$이 성립되어야 한다. 그러므로, 이러한 α_i들의 집합의 차원은 $m-o$이 되어야 하고 x_i들에 대한 해집합의 차원도 $m-o$이 되어야 한다. 위의 알고리즘의 계산복잡도는 결국 $\dim(V) \geq 1$가 되게 하는 $\lambda_i \in F_q$의 선택의 확률에 달려있다. 그런데 $\dim(V) \geq 1$이 되기 위해서는 $\alpha_{m-o+1} = ... = \alpha_m = 0$가 되어야 하고 s_{ij}는 가역행렬 M_S의 원소이며 $\alpha_j = \sum_{i=1}^{m} \lambda_i s_{ij}$를 만족하기 때문에 만족하는 λ_i들의 집합의 차원도 $m-o$가 되어야 한다. 따라서 $\dim(V) \geq 1$이 되기 위한 $\lambda_i \in F_q$들이 선택될 확률은 $\frac{q^{m-o}}{q^m} = 1/q^o$이 된다. 따라서 위의 HighRank 알고리즘의 계산복잡도는 $O(q^o)$정도가 된다.

정리. 공개키의 각 컴포넌트함수의 행렬표현 중에 $u+1$개를 임의로 선택하여 만들어진

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

일차결합이 마지막 변수를 제외한 다른 변수가 0인 되는 벡터들의 집합을 포함하는 kernel을 가질 때 이러한 kernel은 위의 알고리즘을 사용하여 $O(q^u \cdot (un^2+n^3/6))$정도의 유한체 곱셈을 통해서 이 kernel을 찾을 수 있다.
따라서 Rainbow 전자서명의 마지막 계층에서의 vinegar 변수를 v, oil 변수를 o라 한다고 하면 위의 정리에 의해서 HighRank 공격의 복잡도는 다음과 같다.

$$C_{\text{HighRank}}(q,v,o,n) = q^o \cdot \frac{n^3}{6}$$

3. MQQ-SIG 안전성 분석

2008년 Gligoroski등이 Multivarite Quadratic Quasigroup(MQQ)라는 새로운 개념을 이용하여 전자서명 알고리즘을 제안하였다[GMK08]. 하지만, 이 알고리즘은 Gröbner basis를 이용한 direct attack에도 쉽게 깨졌다[MDBW09]. Gligoroski 등은 이러한 단점을 보완하기 위해 공개키의 상당부분을 감추는 마이너스 아이디어를 이용한 새로운 서명 알고리즘[GOJP12]과 새로운 암호화 알고리즘[GS12]을 제안하였다. 이 알고리즘들의 장점은 같은 안전성을 갖는 다른 서명 및 암호화 알고리즘에 비해 굉장히 빠르다는 것이었다.

하지만, 이들 스킴의 중앙 비밀 함수인 F는 아주 작은 수의 2차항만을 가지고 있었으며, MQQ의 특성을 이용하여 마지막 변수인 x_n에 관한 2차항이 아예 없도록 변형을 할 수 있었다. 결국 이들 스킴들은 키 복구 공격에 다항식 시간 안에 깨지게 되었으며[FGPS15], 따라서 MQQ를 이용하여 암호시스템을 만드는 것은 어렵게 되었다.

MQQ에 관한 기본적인 사실이나 MQQ-SIG에 관해서는 2장 2절 1. 나. (4)에서 이미 소개되어있다. 그 이후에 설계된 MQQ-ENC의 설계 원리도 MQQ-SIG와 굉장히 유사하다. 다만, MQQ-SIG와 달리 이는 암호화 알고리즘이기 때문에 복호화가 가능하도록 공개키를 절반이나 지워버렸던 것을 고정된 작은 수 r에 대해 r개만 지우는 것으로 설계하였고, 대신 복호화가 가능하도록 해쉬함수를 사용하였다. 또한 이 설계된 스킴은 F_{q^k} 형태의 유한체에서도 정의가 가능하며, left quasigroup을 사용하였다. 여기서 사용된 MQQ는 아래와 같이 정의되었다.

$$\overline{q_u}(x,y) := p_u(y_u) + \sum_{i=1}^{d}\sum_{j=1}^{d}\alpha_{i,j}^{(u)}x_ix_j + \sum_{s<i,j\le d}\beta_{i,j}^{(u)}y_iy_j + \sum_{i=1}^{d}\sum_{j=u+1}^{d}\gamma_{i,j}^{(u)}x_iy_j + \sum_{i=1}^{d}\delta_i^{(u)}x_i + \sum_{i=u+1}^{d}\epsilon_i^{(u)}y_i + \eta^{(u)}$$

여기서 p가 짝수일 때는 $p_u(y_u) = a^{(u)}y_u$ or $a^{(u)}y_u^2$ 이고 홀수일 때는 $p_u(y_u) = a^{(u)}y_u$이며 $a^{(u)} \neq 0$이다. 함수 $\bar{q} = (\overline{q_1}, \overline{q_2}, \cdots, \overline{q_d}) : F_{q^k}^{2d} \mapsto F_{q^k}^{d}$를 위수가 q^{kd}인 LMQQ(left multivariate quadratic quasigroup)으로 부른다.

이 때, 두 가역인 d차 정사각행렬 D, D_y와 $c, c_y \in F_q^d$에 대해 $\hat{q}(x,y) = D\bar{q}(x, D_y y + c_y) + c$는

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

위수가 q^{kd}인 LMQQ가 된다. 이 때 $\hat{q}$는 $\bar{q}$에 linear isotopic이라고 한다. 추천된 파라미터(안전성 레벨 2^{128})는 $d=8, q=2$, $(n,k,r)=(256,1,8)$, $(128,2,4)$, $(64,4,4)$, $(32,8,1)$ 이다.

가. 키 복구 공격

MQQ-ENC의 모든 성질들이 MQQ-SIG에 잘 적용되므로, 여기서는 MQQ-ENC에 대한 키 복구 공격을 진행하도록 한다. 위의 LMQQ의 2차항 모양을 보았을 때 중앙 비밀 함수 F의 계수행렬 모양은 그림 6과 같음을 알 수 있다.

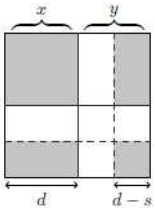


그림 6. $\overline{q_u}$의 2차 형태

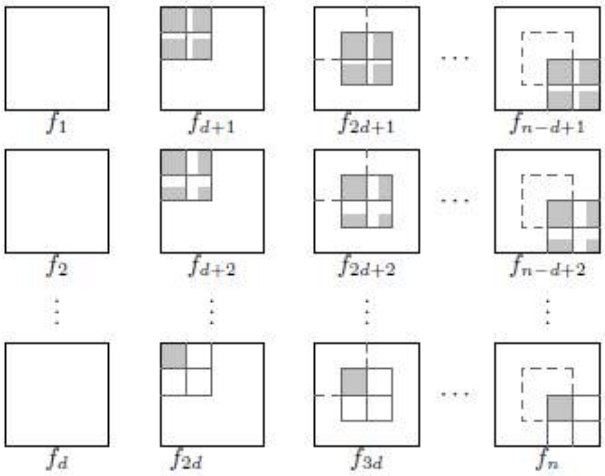


그림 7. MQQ-ENC의 중앙 비밀 함수 F의 2차항의 계수행렬(회색은 임의 값, 흰색은 0)

조금 더 쉽게 설명하고 정렬된 계수 번호를 획득하기 위해 F의 계수 번호는 $x_{n-id+j} \mapsto x_{x-id+d-j+1}$로 변형한다. 이러한 변형은 그림 7의 흰색 밴드를 오른쪽 아래로 놓게 하며 회색 부분이 왼쪽 위에 정사각형의 형태로 뭉치도록 한다.

이제 우리는 n번째 단계에서 $r+2$번째 단계까지 총 $n-r-1$개의 단계를 거칠 것이다. 각 $N \in \{n,\cdots,r+2\}$번째 단계에서 우리는 공개키에서 변수 x_N을 제거하고자 한다. 각 단계의 마지막에서 변수 x_N을 포함하고 있는 유일한 공개 다항식이 제거될 것이며, 이것은 특정한 형태의 good key 쌍 $(\widetilde{S}_N, \widetilde{T}_N)$를 찾아 공개키 함수의 양쪽에 $\widetilde{S}_N \circ P \circ \widetilde{T}_N$의 형

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

태로 합성하여줌으로써 이루어진다. 최종적인 동치 키의 형태는 $\overline{S'}=\overline{S'_{r+2}}\circ\cdots\circ\overline{S'_n}$과 $\widetilde{T'}=\widetilde{T'_n}\circ\cdots\circ\widetilde{T'_{r+2}}$이다.

먼저 n번째 단계에서는 윗 단락에서의 변수의 변환으로 x_n은 어떠한 중앙 비밀키 다항식에도 들어있지 않다. 따라서, [TW12b]에서 사용한 방법과 비슷하게 good key를 찾을 수 있다. 그런데 x_n이 어떤 비밀키에도 들어가 있지 않기 때문에 이번 단계에서는 $\Sigma=S$로 선택하여 $S_n=I$로 잡을 수 있다.Ω의 경우 Rainbow에서의 Ω와 동일한 형태로 잡으면 그림 8과 같은 형태의 $\widetilde{T'_n}$를 얻을 수 있다.

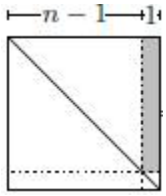


그림 8. $\widetilde{T'_n}$의 모양

그림 8과 같은 모양의 $\widetilde{T'_n}$은 만약 T_{nn}의 값이 0이라면 존재할 수 없다. 따라서 이러한 good key가 존재할 가능성은 $1-\frac{1}{q}$이다. 이 모양의 $\widetilde{T'_n}$를 찾기 위해서는 $m(n-1)$개의 식과 $n-1$개의 변수로 이루어진 1차 연립방정식 시스템을 찾는 것으로 충분하다.

나머지 단계들에서는 N번째 단계에서는 x_N이 F의 다항식 중 1개에만 들어있게 된다. 나머지는 전 단계에서 모두 제거된 상태이기 때문이다. 따라서 x_N을 제거하기 위해서는 공개키 중 두 개의 다항식의 1차결합 만으로도 충분하다. 이것을 $P^{(1)},P^{(k)}$라고 하자. 변수의 개수가 N개이고 이를 제거한 후의 변수는 $N-1<N$이므로, 이는 $Rank(P^{(k)}+\lambda P^{(1)})<N$의 MinRank 문제처럼 보인다. 이것의 해를 찾기 위해서는 그림 9와 같은 모양의 good key를 찾아야 한다.

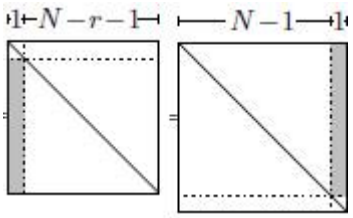


그림 9. good key $\widetilde{S'_N}$(왼쪽), $\widetilde{T'_N}$(오른쪽)의 모양

위 과정의 이론적인 복잡도는 $O(n^{3\alpha+1})$로, 성공확률은 $(1-\frac{1}{q})(1-\frac{1}{q^{n-3}})$이다. 여기서 $2\le\alpha<3$는 선형대수학 상수이다. 하지만, 실질적인 복잡도는 그보다 낮다. 이 공격의 구

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

현결과, 2^{128}의 안전성을 갖는 MQQ-ENC의 키 복구 공격의 이론적 복잡도는 $2^{59}(\alpha=3)$였는데 실질적인 구현 결과 2^{51}의 복잡도를 가지며 동치 키를 찾아내는데 9.1일이 걸렸다. 또한 2^{80}의 안전성을 갖는 MQQ-SIG의 키 복구 공격의 이론적 복잡도는 $2^{62}(\alpha=3)$이었는데 실질적인 구현 결과 2^{48}의 복잡도를 가지며 동치 키를 찾아내는데 1.4일이 걸렸다.

이러한 이론 및 실험결과는 MQQ 기반 스킴은 모두 깨졌음을 의미하고, MQQ 구조를 이용하여 공격을 시도한만큼 이제 MQQ 구조를 사용하여 안전한 암호 스킴을 설계하기는 아주 어려워졌고, 아마 quasigroup 이론에 대한 깊은 통찰이 요구될 것으로 보인다.

4. Simple Matrix Encryption Scheme 안전성 분석

앞의 38~40쪽에서 언급한 TSMES에 대한 공격은 ABC 암호화 스킴의 Tensor 곱 버전 이외의 다른 버전에는 적용되지 않는다. 하지만, 아주 최근에 ABC 암호화 스킴 전체를 깰 수 있는 공격 방법이 제안되었다[Gu16]. 여기서는 이 공격방법을 소개하도록 한다.

우선 이 공격은 $P=S\circ F\circ T$로부터 $S^{-1}\circ P=F\circ T$를 유도한다. 따라서 어떤 메시지 $M=(M_1,\cdots,M_n)$에 대응하는 암호문 $C=(C_1,\cdots,C_m)$에 대해 $C=P(M)=S(F(T(M)))$으로부터 $S^{-1}(C)=F(T(M))$이 성립한다. 이제, S^{-1}, T의 계수행렬을 아래와 같이 잡고,

$$S^{-1}=\begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,m} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,m} \end{pmatrix},\quad T=\begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n} \end{pmatrix}$$

$S^{-1}(C)=\overline{Y}=(\overline{y_1},\overline{y_2},\cdots,\overline{y_m})\in F_q^m$, $T(M)=\overline{X}=(\overline{x_1},\overline{x_2},\cdots,\overline{x_n})\in F_q^n$로 잡으면 $\overline{y_i}=\sum_{j=1}^{m} s_{i,j}C_j$ $(1\le i\le m)$, $\overline{x_i}=\sum_{j=1}^{n} t_{i,j}M_j (1\le i\le n)$를 만족한다. A, B, C의 모양이 아래와 같으므로,

$$A=\begin{pmatrix} x_1 & x_2 & \cdots & x_s \\ x_{s+1} & x_{s+2} & \cdots & x_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ x_{s(s-1)+1} & x_{s(s-1)+2} & \cdots & x_n \end{pmatrix}, B=\begin{pmatrix} b_1 & b_2 & \cdots & b_s \\ b_{s+1} & b_{s+2} & \cdots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ b_{s(s-1)+1} & b_{s(s-1)+2} & \cdots & b_n \end{pmatrix}, C=\begin{pmatrix} c_1 & c_2 & \cdots & c_s \\ c_{s+1} & c_{s+2} & \cdots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{s(s-1)+1} & c_{s(s-1)+2} & \cdots & c_n \end{pmatrix}$$

이때, $b_i=\sum_{j=1}^{n} b_{i,j}x_j$, $c_i=\sum_{j=1}^{n} c_{i,j}x_j$로 정의하면, ABC의 복호화 과정에서의 $\overline{A}, \overline{B}, \overline{C}$는 아래와 같다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

$$\overline{A}=\begin{pmatrix} \overline{x_1} & \overline{x_2} & \cdots & \overline{x_s} \\ \overline{x_{s+1}} & \overline{x_{s+2}} & \cdots & \overline{x_{2s}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{x_{s(s-1)+1}} & \overline{x_{s(s-1)+2}} & \cdots & \overline{x_n} \end{pmatrix}, \overline{B}=\begin{pmatrix} \sum_{j=1}^{n} b_{1,j}\overline{x_j} & \sum_{j=1}^{n} b_{2,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} b_{s,j}\overline{x_j} \\ \sum_{j=1}^{n} b_{s+1,j}\overline{x_j} & \sum_{j=1}^{n} b_{s+2,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} b_{2s,j}\overline{x_j} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{n} b_{s(s-1)+1,j}\overline{x_j} & \sum_{j=1}^{n} b_{s(s-1)+2,,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} b_{n,j}\overline{x_j} \end{pmatrix}$$

$$\overline{C}=\begin{pmatrix} \sum_{j=1}^{n} c_{1,j}\overline{x_j} & \sum_{j=1}^{n} c_{2,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} c_{s,j}\overline{x_j} \\ \sum_{j=1}^{n} c_{s+1,j}\overline{x_j} & \sum_{j=1}^{n} c_{s+2,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} c_{2s,j}\overline{x_j} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{n} c_{s(s-1)+1,j}\overline{x_j} & \sum_{j=1}^{n} c_{s(s-1)+2,,j}\overline{x_j} & \cdots & \sum_{j=1}^{n} c_{n,j}\overline{x_j} \end{pmatrix}$$

한편 ABC의 복호화 과정에서 $\overline{Y}$의 각 성분을 s개에 한 행씩 차례로 나열하여 두 개의 s차 정사각행렬을 만든 것을 $\overline{E_1}, \overline{E_2}$로 부르기로 하였다. 즉, $\overline{E_1}, \overline{E_2}$는 아래와 같다.

$$\overline{E_1}=\begin{pmatrix} \overline{y_1} & \overline{y_2} & \cdots & \overline{y_s} \\ \overline{y_{s+1}} & \overline{y_{s+2}} & \cdots & \overline{y_{2s}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{y_{s(s-1)+1}} & \overline{y_{s(s-1)+2}} & \cdots & \overline{y_n} \end{pmatrix}, \overline{E_2}=\begin{pmatrix} \overline{y_{n+1}} & \overline{y_{n+2}} & \cdots & \overline{y_{n+s}} \\ \overline{y_{n+s+1}} & \overline{y_{n+s+2}} & \cdots & \overline{y_{n+2s}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{y_{n+s(s-1)+1}} & \overline{y_{n+s(s-1)+2}} & \cdots & \overline{y_m} \end{pmatrix}$$

이제 우리는 $\overline{E_1}=\overline{A}\overline{B}, \overline{E_2}=\overline{A}\overline{C}$로부터 m개의 식을 얻는다. 각 식에 $\overline{y_i}=\sum_{j=1}^{m} s_{i,j}C_j$ $(1 \le i \le m)$, $\overline{x_i}=\sum_{j=1}^{n} t_{i,j}M_j (1 \le i \le n)$를 대입하면, 식 m개와 총 변수의 개수는 $b_{i,j}, c_{i,j}, t_{i,j} (1 \le i,j \le n)$, $s_{i,j} (1 \le i,j \le m)$의 $3n^2+m^2=7n^2$개의 변수로 이루어진 3차식 시스템을 한 개 얻는다. 이제 $\overline{E_1}$의 1행 1열의 성분으로부터 얻은 식을 살펴보자.

$$\overline{y_1}=\sum_{k=1}^{s}\overline{x_k}\cdot\sum_{j=1}^{n} b_{ks-(s-1),j}\overline{x_j}$$
$$\sum_{j=1}^{m} s_{1,j}C_j=\sum_{k=1}^{s}\sum_{i_1=1}^{n} t_{k,i_1}M_{i_1}\cdot\sum_{j=1}^{n} b_{ks-(s-1),j}\sum_{i_2=1}^{n} t_{j,i_2}M_{i_2}$$
$$\sum_{k=1}^{m} s_{1,k}C_k=\sum_{j=1}^{s}\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n} t_{j,i_1}b_{js-(s-1),i_3}t_{i_3,i_2}M_{i_1}M_{i_2}$$

위의 형태이다. M_i, C_j는 메시지와 암호문에서 나온 식이므로, 식 1개에 총 sn^3개의 3차항이 나올 수 있다. 식이 총 m개이므로 3차항의 개수는 총 $2sn^4$개다. 한편 $s_{i,j}$는 총 m^2

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

개이므로, 위의 3차항들을 모두 일차화하였을 경우 총 변수가 $2sn^4+m^2$개이고 m개인 일차방정식 시스템을 얻는다. 한편, 우리는 공개키를 알 수 있으므로 이를 이용하여 메시지-암호문 쌍을 몇 개든지 얻어낼 수 있다. 메시지-암호문 쌍이 바뀌더라도 식의 구조가 바뀌지는 않기 때문에 우리는 또 다른 메시지-암호문 쌍의 각 성분인 M_i', C_j'를 위에서 얻은 식에 또 대입하여 동일한 변수로 이루어진 또 다른 m개의 식을 얻을 수 있다. 이렇게 계속 진행하여 식의 개수가 변수의 개수보다 많아지게 되면 식과 변수가 같은 1차 연립방정식 시스템을 풀어 해를 구할 수 있다. 이로써 비밀키 B, C, S, T 또는 그것의 동치키를 구할 수 있고 우리는 언제든지 ABC 암호화 스킴으로 암호화된 암호문을 복호화할 수 있다. 이 공격의 복잡도는 식과 변수의 개수가 $2sn^4+m^2$개인 1차 연립방정식 시스템을 푸는 것이므로 아래와 같은 다항식 복잡도를 갖는다.

$$O((2sn^4+m^2)^3\log q) = O((sn^4)^3\log q) = O(s^3n^{12}\log q) = O(n^{13.5}\log q)$$

[TXPD15]의 향상된 ABC 암호화 스킴 또한 비슷한 구조를 가지므로, 동일한 방법을 이용하여 공격할 수 있다. 오히려 원래의 스킴보다 키의 길이가 더 짧기 때문에 공격복잡도는 더 낮아진다. 다만, 저자가 이 공격을 실제로 구현하지 않아 이 공격이 실제로 Simple Matrix Encryption Scheme 계열의 스킴들에 미치는 영향에 대해서 정확히 추측하기는 어렵다. 게다가 ABC 스킴의 저자는 $q=256=2^8, n=64=2^6$으로 잡았는데 이 경우 위 공격의 복잡도는 $O(2^{84})$가 되어 결코 낮은 복잡도가 아니다. 물론 저자는 Gauss 소거법과 연관이 있는 선형대수학 상수($2 \le \alpha < 3$)를 $\alpha=3$으로 잡았으나 실제로는 그보다 훨씬 α값을 낮추는 알고리즘들이 제안되어있기 때문에 실제 공격복잡도는 위에서 언급한 것보다 다소 낮아질 수도 있다. 다만 이 경우에도 n을 키울 경우 이 공격에 대비할 수 있을 것으로 생각된다.

제 2 절. 다변수 이차식 기반 암호알고리즘의 설계 원칙

다변수 이차식 기반 암호알고리즘들은 상대적으로 크기가 큰 유한체에서의 암호알고리즘과 작은 유한체에서의 암호알고리즘으로 나눌 수 있다. 큰 유한체에서의 암호알고리즘은 작은 크기의 유한체에서 수십정도의 확장차수를 가지는 유한확장체에서의 비밀함수를 사용한다. 이러한 비밀함수는 대부분 일대일 함수여서 암호알고리즘과 전자서명을 만드는데 사용되고 있다. 이에 반해 작은 유한체에서의 다변수 이차식 기반 암호알고리즘은 작은 크기의 유한체에서 정의된 비밀함수를 사용하고 있으나, 아직까지 암호함수로 사용될 수 있는 비밀함수는 알려져 있지 않고, 단지 전자서명으로 사용될 수 있는 함수만이 제안되고 있다. 이러한 암호알고리즘으로 대표적인 것이 tame transformation을 사용하는 TTS와 oil과 vinegar라는 두 가지 종류의 변수를 가지는 UOV와 Rainbow 전자서명시스템이 있다. 작은 유한체에서의 다변수 이차식 기반 암호알고리즘은 큰 유한체에서의 암호알고리즘과 다르게 안전성을 위해서 비밀함수는 상대적으로 많은 이차항들을 가져야 한

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

다. 본 절에서는 작은 유한체에서의 다변수 이차식 기반 암호알고리즘의 비밀함수가 안전하기 위한 몇 가지 조건들에 대해서 살펴본다.

1. MinRank (Low rank 공격) 공격에 안전하기 위한 조건

유한체 F_q에서 정의된 비밀함수 $F=(f^{(1)}(x_1,...,x_n),\cdots,f^{(m)}(x_1,...,x_n)):F_q^n \rightarrow F_q^m$이 있을 때, 다변수 이차다항식 $f^{(i)}(x_1,...,x_n)$는 대칭행렬 M_i에 대해서 $(x_1,...,x_n)\cdot M_i\cdot(x_1,...,x_n)^t$로 표현될 수 있다. 그러면 TTS와 Rainbow 암호시스템들은 앞부분의 컴포넌트 함수 $f^{(i)}$들이 뒷부분의 컴포넌트 함수들보다 작은 rank를 가지게 되는데, low rank 공격은 MinRank를 가지는 비밀함수의 컴포넌트 함수를 찾고 이를 이용하여 비밀함수를 숨기는 아핀변환 S와 T, 최종적으로 비밀함수 F를 찾는 공격이다.

Low rank 공격은 Goubin과 Courtois가 TPM 암호시스템을 공격하기 위하여 처음 제안한 것으로 MinRank 알고리즘이라 불린다. 이것은 다음과 같다.[CSV94]

MinRank 알고리즘	
입력:	자연수 r, 행렬 $M_i \in F_q^{n\times n}$ for $i=1,...,m$ (여기서 M_i는 공개키 함수의 각 컴포넌트에 있는 이차다변수 다항식의 행렬 표현)
출력:	$\lambda_i \in F_q$ for $i=1,...,m$ $s.t.$ $rank(\sum_{i=1}^{m}\lambda_i M_i) \le r$
1:	k개의 랜덤한 벡터 $X_i \in F_q^n$을 선택한다. (여기서 $k=\left\lceil \frac{m}{n} \right\rceil$로 선택한다.)
2:	변수 λ_i들에 대하여 행렬 $\sum_{i=1}^{m}\lambda_i M_i$를 M이라고 놓는다.
3:	$i=1,...,k$에 대하여 $M\cdot X_i=0$를 만족하는 λ_i를 구한다.

표 20. MinRank 알고리즘

위의 알고리즘에서 미지수 λ_i가 n개 있으며, 과정 3에서 $i=1,...,k$에 대하여 $M\cdot X_i=0$에 의해서 식이 kn개가 생기므로 결국 위의 알고리즘은 MinRank 문제를 n개의 미지수와 kn개의 식으로 이루어진 일차 연립방정식을 푸는 문제로 바꾸게 된다. 이러한 연립방정식은 $k=m/n$으로 택하였기 때문에 만약 해가 존재하게 된다면 그것은 유일하게 되고 따라서 비밀함수의 가장 작은 rank를 가지는 첫 번째 컴포넌트 함수를 찾게 된다. 이러한 연립방정식은 Gauss 소거법을 이용하여 풀면 약 $m^2(nk/2-m/6)+mn^2k)$번의 유한체 곱셈으로 풀 수 있다. 그러나 MinRank 알고리즘의 가장 큰 어려움은 과정 1에서 랜덤한 벡터 $X_i \in F_q^n$를 선택하는 과정이다. 왜냐하면 $M\cdot X_i=0$를 만족하기 위해서는 랜덤하게 선

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

택한 벡터 X_i가 $\ker(M)$에 들어가야 하기 때문이다. 알고리즘에서 찾기 원하는 행렬이 $rank(M) < r$을 만족하는 행렬이므로 이런 행렬 M에 대하여 $\ker(M)$은 $n-r$의 차원을 가지게 된다. 따라서 F_q^n의 원소를 랜덤하게 선택했을 때, 그 원소가 $\ker(M)$에 들어갈 확률은 $q^{n-r}/q^n = 1/q^r$이 된다. 이러한 벡터를 k개를 선택해야 하므로 이 벡터들이 모두 $\ker(M)$에 들어갈 확률은 $1/q^{kr}$이 된다. 따라서 MinRank 알고리즘의 복잡도는 $O(q^{kr} \cdot (m^2(nk/2-m/6)+mn^2k))$이 된다.

정리 7. 비밀키의 형태와 상관없이 공개키의 각 컴포넌트함수의 행렬표현들이 최소 랭크 r을 갖는 행렬들의 일차결합 $M=\sum_{i=1}^{m}\lambda_i M_i$이 존재한다면 위의 알고리즘은 $O(q^{kr} \cdot m^2(nk/2-m/6)+mn^2k))$정도의 유한체 곱셈을 통해서 $\ker M$에 들어가는 벡터를 찾을 수 있다.

2. High rank 공격에 안전하기 위한 조건

Coppersmith 등에 의해서 제안된 HighRank 공격[CSV94]은 MinRank 공격의 반대공격이라고 볼 수 있다. MinRank 공격은 작은 rank를 가지는 행렬들을 찾는 공격으로 이러한 행렬들은 반대로 큰 차원의 kernel을 가지게 된다. 따라서 MinRank 공격은 작은 수의 layer에 의해서 공유된 큰 kernel을 찾는 공격이라고 할 수 있다. 큰 kernel이란 0의 값을 가지는 변수들이 적은 변수들의 집합을 의미한다. 따라서 작은 수의 layer에 의해서 공유된 큰 kernel을 찾는다는 것은 적은 수의 변수가 사용된 컴포넌트 함수를 찾는다는 것을 의미한다. 반면에 HighRank 공격은 큰 rank를 가지는 행렬들을 찾는 공격이다. MQ-PKC의 공개키의 컴포넌트 함수의 이차항들의 행렬은 아핀 변환에 의해서 서로 섞이게 되므로 대부분은 full rank를 가지게 된다. 따라서 이러한 행렬들은 일차결합을 하면 많은 경우에 high rank를 가지게 된다. 행렬이 큰 rank를 가지게 되면 작은 차원의 kernel을 가지게 되므로 결국 HighRank 공격은 MinRank 공격과는 반대로 많은 수의 layer에 의해서 공유된 작은 kernel을 찾는 공격이라고 할 수 있다. 즉, 적은 수의 layer에서 사용된 변수를 찾는 공격이라고 할 수 있다. HighRank 공격의 아이디어는 비밀키를 이루는 이차 다변수 다항식에서 일부의 변수들은 마지막 layer에 있는 다항식에만 사용된다는 점이다. 즉, 마지막 전의 layer들에서 이러한 변수들이 사용되지 않기 때문에 이러한 layer에서의 다변수 이차 다항식의 행렬들은 full rank를 가지지 않게 된다. 따라서 만약에 이렇게 마지막 layer에서만 사용되는 변수들이 있다면 HighRank 공격을 통해서 마지막 전의 layer를 분리해 낼 수 있는 것이다. m개의 $n \times n$ 행렬F_i은 다음을 만족한다고 하자.

- $rank(F_i) \le rank(F_{i+1})$
- $rank(F_{m-o}) < rank(F_{m-o+1}) = \ldots = rank(F_m) = n$

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

또한 가역행렬 M_T에 대해서 행렬 P_i가 다음과 같이 정의되었다고 하자.

$$P_i = \sum_{j=1}^{m} s_{ij}(M_T^t \cdot F_i \cdot M_T)$$

그러면 다음의 HighRank 알고리즘은 $M_T^t \cdot F_{m-1} \cdot M_T$와 같은 rank를 갖는 P_i들의 일차결합을 찾는 알고리즘이다.

HighRank 알고리즘	
입력:	행렬 $P_i \in F_q^{n \times n}$ for $i = 1,...,m$
출력:	$\lambda_i \in F_q$ for $i = 1,...,m$ $s.t.$ $rank(\sum_{i=1}^{m} \lambda_i P_i) = rank(M_T^t \cdot F_{m-1} \cdot M_T)$
1:	m개의 랜덤한 원소 $\lambda_i \in F_q$을 선택한다.
2:	λ_i들에 대하여 행렬 $\sum_{i=1}^{m} \lambda_i P_i$를 P이라고 놓고 $V = \ker(P)$를 계산한다.
3:	만약 $\dim(V) \geq 1$이면 $(\sum_{i=1}^{m} x_i P_i)V = 0$을 만족하는 해 x_i를 구한다.
4:	만약 해집합의 차원이 $m-o$면 $\lambda_i \in F_q$ for $i = 1,...,m$

표 21. HighRank 알고리즘

위의 알고리즘에 대한 설명은 다음과 같다. 먼저 V를 다시 계산해 보면,

$$\begin{aligned} P &= \sum_{i=1}^{m} \lambda_i P_i = \sum_{i=1}^{m} \lambda_i \sum_{j=1}^{m} s_{ij} M_T^T \cdot F_j \cdot M_T \\ &= M_T^T \cdot (\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{m} s_{ij} F_j) \cdot M_T \\ &= M_T^T \cdot (\sum_{j=1}^{m} \alpha_j F_j) \cdot M_T \quad where \quad \alpha_j = \sum_{i=1}^{m} \lambda_i s_{ij} \end{aligned}$$

여기서 M_T는 가역행렬이므로 $rank(P) = rank(\sum_{j=1}^{m} \alpha_j F_j)$임을 알 수 있다. 따라서 $\dim(V) = n - rank(\sum_{j=1}^{m} \alpha_j F_j)$가 된다. 그런데 $rank(F_{m-o+1}) = ... = rank(F_m) = nrank(F_m) = n$이므로 $\dim(V) \geq 1$가 되기 위해서는 $\alpha_{m-o+1} = ... = \alpha_m = 0$이 되야 한다. 그러므로, 이러한 α_i들의 집합의 차원은 $m-o$이 되어야 하고 x_i들에 대한 해집합의 차원도 $m-o$이 되어야 한다. 위의 알고리즘의 계산복잡도는 결국 $\dim(V) \geq 1$가 되게 하는 $\lambda_i \in F_q$의 선택의 확률에 달려있다. 그런데 $\dim(V) \geq 1$이 되기 위해서는 $\alpha_{m-o+1} = ... = \alpha_m = 0$가 되어야 하고

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

s_{ij}는 가역행렬 M_S의 원소이며 $\alpha_j = \sum_{i=1}^{m} \lambda_i s_{ij}$를 만족하기 때문에 만족하는 λ_i들의 집합의 차원도 $m-o$가 되어야 한다. 따라서 $\dim(V) \geq 1$이 되기 위한 $\lambda_i \in F_q$들이 선택될 확률은 $\frac{q^{m-o}}{q^m} = 1/q^o$이 된다. 따라서 위의 HighRank 알고리즘의 계산복잡도는 $O(q^o)$정도가 된다.

정리 8. 공개키의 각 컴포넌트함수의 행렬표현중에 $u+1$개를 임의로 선택하여 만들어진 일차결합이 마지막 변수를 제외한 다른 변수가 0인 되는 벡터들의 집합을 포함하는 kernel을 가질 때 이러한 kernel은 위의 알고리즘을 사용하여 $O(q^u \cdot (un^2 + n^3/6))$정도의 유한체 곱셈을 통해서 이 kernel을 찾을 수 있다.

3. UOV 공격에 안전하기 위한 조건

Unbalanced Oil-and-Vinegar(UOV) 공격[KS98]은 다변수 이차식 기반 암호시스템에 사용되는 변수를 O와 V의 두 집합으로 나눌 수 있고, O에 속하는 변수들로 이루어진 이차항이 나타나지 않을 때 이루어질 수 있다. O에 속하는 변수를 oil 변수, V에 속하는 변수를 vinegar 변수라고 하자. 또한 $|V| = v, |O| = o, n = v+o$라고 정의하면 UOV 공격은 다음과 같다.

UOV 공격	
입력:	행렬 $P_i \in F_q^{n \times n}$ for $i = 1,...,m, O = \{x = (x_1,...,x_n) \in F_q^n \mid x_1 = ... = x_v = 0\}$
출력:	$M_T^{-1} \cdot O$의 기저
1:	m개의 랜덤한 원소 $\lambda_i \in F_q$을 선택한다.
2:	λ_i들에 대하여 행렬 $\sum_{i=1}^{m} \lambda_i P_i$를 W이라고 놓고 가역행렬인 P_j에 대하여 $\widehat{W} = P_j^{-1} \cdot W$를 계산한다.
3:	$\widehat{W}$의 invariant 부분공간을 구한후, 각 부분공간이 $M_T^{-1} \cdot O$의 부분공간인지를 확인한다.
4:	$M_T^{-1} \cdot O$의 기저를 찾을때까지 위의 과정을 반복한다.

표 22. UOV 공격 알고리즘

UOV 공격의 핵심 아이디어는 비밀함수의 모든 컴포넌트 함수에 O에 속하는 변수들로 이루어진 이차항이 나타나지 않기 때문에 임의의 두 개의 컴포넌트 함수 F_i, F_j에 대하여 $F_i^{-1}(F_j(O))$가 일정확률로 다시 O의 부분공간이 된다는 것을 이용하는 것이다. 물론 비밀

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

함수 F_i, F_j는 모르기 때문에 대신에 공개키의 컴포넌트 함수를 이용하여 불변공간을 찾으면 이것이 $M_T^{-1} \cdot O$의 부분공간이 되기 때문에 이로부터 M_T에 해당하는 비밀 affine 변환 T를 찾아낼 수 있다는 것이다. $F_i^{-1}(F_j(O))$가 다시 O의 부분공간이 될 확률은 다음과 같다.

보조정리 8. 역행렬이 존재하는 F_i에 대하여 $F_i^{-1}(F_j(O))$가 O의 부분공간이 될 확률은 $1/q^{(2v-n-1)}$보다 크다.

(증명) $F_i^{-1}F_j$를 ϕ라 하자. 그러면 다음을 알 수 있다.

$$\dim(O) = \dim(\phi(O)) = o$$
$$\dim(V) = \dim(\phi^{-1}(V)) = v$$

이 사실과 $\phi(O) \subseteq \phi^{-1}(V)$로부터 0이 아닌 O의 원소 α에 대해서 다음의 확률을 계산할 수 있다.

$$\Pr ob(\phi(\alpha) = c \cdot \alpha) = \frac{q-1}{q^v-1}$$

$\phi(\alpha) = c \cdot \alpha$이면 임의의 상수 d에 대해 $\phi(d \cdot \alpha) = c \cdot (d \cdot \alpha)$가 성립하기 때문에 다음의 확률을 얻을 수 있다.

$$\begin{aligned}\Pr ob(\phi(O) \subseteq O) &\geq \Pr ob(\phi(\alpha) = c \cdot \alpha \mid \alpha \in O) \\ &= \sum_{\alpha \in O} \Pr ob(\phi(\alpha) = c \cdot \alpha) \\ &= \frac{(q^o-1)(q-1)}{(q-1)(q^v-1)} \approx q^{o-v-1}\end{aligned}$$

■

위의 알고리즘의 스텝 3에서 $M_T^{-1} \cdot O$의 부분공간인지의 확인은 다음 사실을 이용하여 이루어질 수 있다.

보조정리 9. 만약 $H \subset M_T^{-1} \cdot O$이면 H의 모든 원소 x,y에 대해서 다음이 성립한다.

$$x^t \cdot P_i \cdot y = 0$$

지금까지의 UOV 공격에 관한 보조정리와 알고리즘을 정리하면 다음과 같다.

정리 9. 만약 다변수 이차식 기반 전자서명의 공개키 함수가 n개의 변수와 m개의 컴포넌트 다항식을 가지고, 이러한 다항식들이 최소 v개의 vinegar 변수를 가진다면 전자서명은 $q^{(2v-n-1)}(n-v)^4$번 정도의 유한체 곱셈으로 위조될 수 있다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

4. Tame을 기반으로 한 다변수 이차식 기반 암호시스템의 안전한 설계를 위한 조건

앞에서 설명한 Rank 공격과 UOV 공격이외의 일반적인 공격에는 groebner 기저를 이용하여 직접 다변수 이차 연립방정식을 푸는 direct 공격과 linearization 공격이 있다. 앞 절의 내용을 정리하여 tame을 기반으로 한 다변수 이차식 기반 암호시스템의 안전한 설계를 위한 조건들은 다음과 같다.

정리 10. Tame 구조를 기반으로 하고 n개의 변수와 m개의 컴포넌트의 공개키를 가지는 다변수 이차식 기반 암호시스템이 C-비트의 안전도를 가지기 위해서는 다음의 조건들을 고려해야 한다.

1. 비밀함수의 각 컴포넌트 다항식에 나타나는 $x_i x_j (i \neq j)$와 같은 항들(cross-term)의 최소개수를 l이라 할 때, 파라미터는 다음 조건을 만족해야 한다.

$$q^{2l} \cdot m^2(nk/2 - m/6) + mn^2k)) \geq C$$

2. 비밀함수에 사용된 변수가 최소한 u개의 컴포넌트 다항식에 사용된다면 파라미터는 다음 조건을 만족해야 한다.

$$q^u \cdot (un^2 + n^3/6) \geq C$$

3. 비밀함수의 모든 cross 항에 최소한 한 개의 변수가 속한 인덱스의 집합의 최소 크기를 v라 하면 파라미터는 다음 조건을 만족해야 한다.

$$q^{(2v-n-1)}(n-v)^4 \geq C$$

4. c_0, c_1, γ와 $d_{reg} = \min\left\{D \mid \frac{(1-t^2)^m}{(1-t)^{n-k}}\right.$에 나타나는 t^D항의 계수가 0보다 작거나 같다.$\left.\right\}$, $T = \binom{n-k+d_{reg}-1}{d_{reg}}$, 일차연립방정식을 푸는 선형대수 상수 w에 대해서 파라미터는 다음 조건을 만족해야 한다.

$$\min_k q^k \cdot m^\gamma T^w (c_0 + c_1 \log T) \geq C$$

5. 비밀함수는 부분방정식으로 over-determined 연립방정식을 가지면 안 된다.

(증명) 1에서 3번의 사실은 앞 절의 low rank, high rank, UOV 공격에 의해서 성립함을 알 수 있다. 4번의 사실은 gröbner 기저를 효율적으로 계산할 수 있는 HybridF_5 알고리즘[BFP08]을 사용하여 n개의 변수와 m개의 이차 다변수 연립방정식을 푸는 계산량이다. 따라서 4번의 조건을 만족해야 direct 공격을 피할 수 있다. 5번의 사실은 만약 비밀함수는 부분방정식으로 over-determined 연립방정식을 가지면 XL 타입의 공격이 매우 작은 계산량을 가지고 이루어질 수 있다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 3 절. 다변수 이차식 기반 알고리즘에 대한 최적 구현

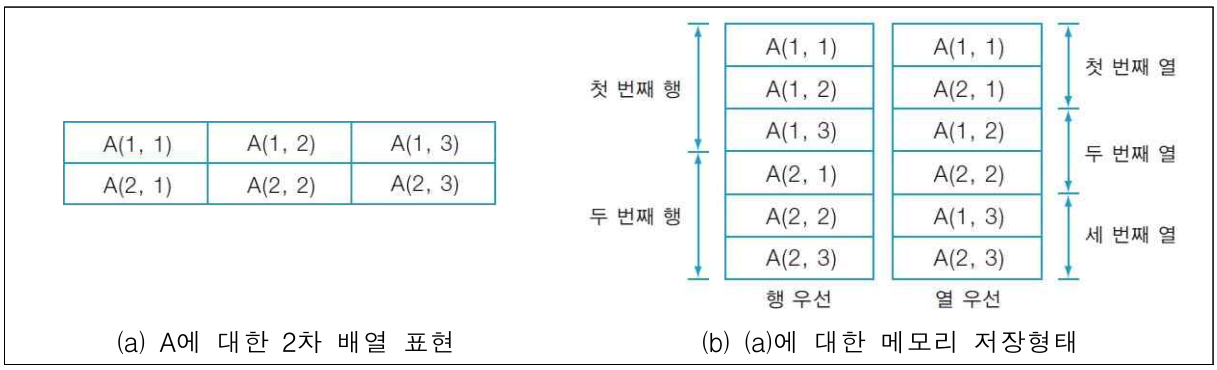
1. 다변수 이차식 기반 알고리즘에 대한 최적화 방안

가. 다항식 표현에 있어 다차원 배열에서 1차원 배열로 변환

1차원 배열 및 메모리 접근 방식 기반의 다변수 이차식 표현 및 접근의 고속화

배열은 동질(homogeneous)의 자료 객체들을 묶고 그룹 이름을 부여하는 자료구조이며, 프로그램을 실행하는 동안 특정한 원소를 상대 주소로 직접 접근할 수 있도록 허용한다. 배열의 요소들은 연속적으로 저장되면 빠르게 접근할 수 있어 실행 시간을 단축한다. 다만, 다변수 이차식을 표현함에 있어 각 계수를 2차 또는 3차의 행렬로 나타낼 수 있는데, 이를 그대로 2차 또는 3차 배열로 처리하면 각 배열 원소를 저장하고 특정 원소에 접근하는데 많은 시간이 소요된다. 이는 2차 또는 3차 배열은 메모리에 저장하기 위해 컴파일러에 의해 1차원 배열로 변환되기 때문이며, 변환 방법은 행 우선, 열 우선, 슬라이스 등 다양하지만 이 중에서 어떤 변환 방법을 적용할지는 프로그램 언어마다 다르다. 예를 들어, 다음 2행 3열의 행렬 A에 대한 2차 배열 표현 및 메모리 저장 형태는 그림 10과 같다.

$$A = \begin{vmatrix} A(1,1) & A(1,2) & A(1,3) \\ A(2,1) & A(2,2) & A(2,3) \end{vmatrix}$$



(a) A에 대한 2차 배열 표현 (b) (a)에 대한 메모리 저장형태

그림 10. 2행 3열 행렬에 대한 2차 배열 표현 및 메모리 저장 형태

C/C++의 컴파일러는 행 우선 방법을 사용하며, 행 우선 방법으로 저장된 2차원 배열 A의 경우 배열 원소 $A[i,j]$에 접근하기 위한 상대 주소는 다음과 같다.

$$\text{base} + ((i - low_1)n_2 + (j - low_2 + 1))w \qquad (1)$$

이때, base는 배열의 시작 주소, i는 행의 인덱스, j는 열의 인덱스, low_1과 low_2는 i와 j

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

의 하한 값이고, n_1은 행 단위의 원소 개수로 i가 취할 수 있는 값의 크기이며, n_2는 열의 크기로 j가 취할 수 있는 값의 크기이다. 즉, $high_2$를 j에 대한 상한 값이라고 하면 $n = high_2 - low_2 + 1$이다. $A[i,j]$의 상대 주소를 계산하는데 컴파일 시간에 i와 j의 값만 모르고 다른 값들을 모두 안다면 위 식 (1)는 식 (2)과 같이 다시 쓸 수 있다. 식 (2)의 뒷부분은 컴파일 시간에 계산할 수 있다.

$$((i*n)+(j+1))*w+\text{base}-((low_1*n)+low_2)*w \qquad (2)$$

행렬을 배열로 처리함에 있어 이러한 1차원 배열, 다차원 배열, 메모리 저장형태, 그리고 상대 주소로의 접근 방식 등의 특성을 고려하여 프로그램을 구현한다면 원하는 원소로 빠르게 접근할 수 있어 실행 시간을 단축할 수 있다. 그래서 최근의 다변수 이차식 기반 암/복호화 또는 서명 알고리즘을 살펴보면, 실행 속도를 더욱 향상시키는 구현 및 실험의 경우 다차원 배열의 사용이 배제되고 1차원 배열 기반의 벡터 사용이 주를 이룬다[AJSP16].

나. 벡터(1차원 배열)의 원소 재구성 및 효과적인 연산 방식

원소 재구성으로 메모리 접근 및 연산 회수 감소

1차원 배열을 사용함에 있어 배열 원소의 구성 및 저장 순서는 특정 원소가 있는 메모리 위치를 찾아가는 상대주소 연산과 해당 배열을 이용하는 전체 연산 횟수를 감소시켜 결과적으로 프로그램의 실행 시간을 줄여주는 효과를 보인다.

5-pass MQDSS 경우에는 F_2 기반 $q=256$ 원소를 갖는 x를 입력 벡터로 사용하는데, 이는 256비트 SIMD 벡터 레지스터 한 개에 딱 맞게 들어간다. 이 256비트 벡터 레지스터의 이용은 AND 연산에 대해 레지스터의 빠른 속도와 병렬 처리를 동시에 사용할 수 있도록 하여 프로그램 전체의 수행 속도를 높여주는데, 그림 11과 같은 원소의 재배열은 벡터 레지스터를 효율적으로 사용하도록 하여 속도 향상의 효과를 더욱 높인다.

&	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	00	11	22	33	04	15	26	37	08	19	2A	3B	0C	1D	2E	3F
	10	21	32	03	14	25	36	07	18	29	3A	0B	1C	2D	3E	0F
	-	-	-	-	44	55	66	77	48	59	6A	7B	4C	5D	6E	7F
	50	61	72	43	54	65	76	47	58	69	7A	4B	5C	6D	7E	4F
	-	-	-	-	-	-	-	-	88	99	AA	BB	8C	9D	AE	BF
	90	A1	B2	83	94	A5	B6	87	98	A9	BA	8B	9C	AD	BE	8F
	-	-	-	-	-	-	-	-	-	-	-	-	CC	DD	EE	FF
	D0	E1	F2	C3	D4	E5	F6	C7	D8	E9	FA	CB	DC	ED	FE	CF
	02	13	-	-	42	53	-	-	82	93	-	-	C2	D3	-	-
	06	17	-	-	46	57	-	-	86	97	-	-	C6	D7	-	-
	0A	1B	-	-	4A	5B	-	-	8A	9B	-	-	CA	DB	-	-
	0E	1F	-	-	4E	5F	-	-	8E	9F	-	-	CE	DF	-	-

그림 11. 효과적인 재배열을 통한 벡터 AND 연산

다변수 이차식 함수인 F 계산은 x의 각 원소들의 페어에 대한 계산으로부터 나온다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

이러한 각 원소들의 페어 계산(bitwise AND 연산)을 8비트 벡터로 살펴보면, 직관적인 계산은 그림 12.(a)와 같은 계산으로 이루어진다.

&	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	-	11	12	13	14	15	16	17
2	-	-	22	23	24	25	26	27
3	-	-	-	33	34	35	36	37
4	-	-	-	-	44	45	46	47
5	-	-	-	-	-	55	56	57
6	-	-	-	-	-	-	66	67
7	-	-	-	-	-	-	-	77

(a) 직관적인 두 벡터간의 AND 연산

&	0	1	2	3	4	5	6	7
	00	11	22	33	44	55	66	77
	10	21	32	43	54	65	76	07
	20	31	42	53	64	75	06	17
	30	41	52	63	74	05	16	27
	40	51	62	73	-	-	-	-

(b) Shift를 이용한 효율적인 AND 연산

그림 12. 8비트 기반 벡터에 대한 효과적인 AND 연산

그림 12의 (a)와 같이 행렬 형태의 직관적으로 계산하는 벡터($q=8$) AND 연산을 (b)와 같은 재배열을 통한 원소의 연산 순서를 바꿔 AND 연산을 수행하면, (b)의 벡터 AND 연산 횟수가 (a)대비 $q/2$회로 감소한다. 이때, 00 11 ... 77에 대한 벡터 연산은 그 결과가 자기자신이므로 별도의 연산이 필요없어 연산 회수에 포함되지 않는다.

2. 고속화를 위한 코드 최적화

코드 최적화는 주어진 코드에 대해 수행 결과는 동일한 의미를 가지면서 실행 시간 또는 메모리를 줄이는 것이다. 최적화는 '효과가 가장 좋다'는 뜻인데, 가장 효과가 좋은 코드를 만들기는 상당히 어려운 일이기 때문에 일반적으로 '기존의 방법보다 계산 회수를 줄이거나 실행 시간을 더 짧게 혹은 기억 용량을 더 적게'라는 의미로 사용한다. 그림 13은 구현 단계별로 적용 가능한 최적화 기법을 보이며, 컴파일러에 의해 최적화되는 단계를 코드 최적화 단계라고 하지만 통상 구현 전 단계에서 코드 최적화는 적용될 수 있다.

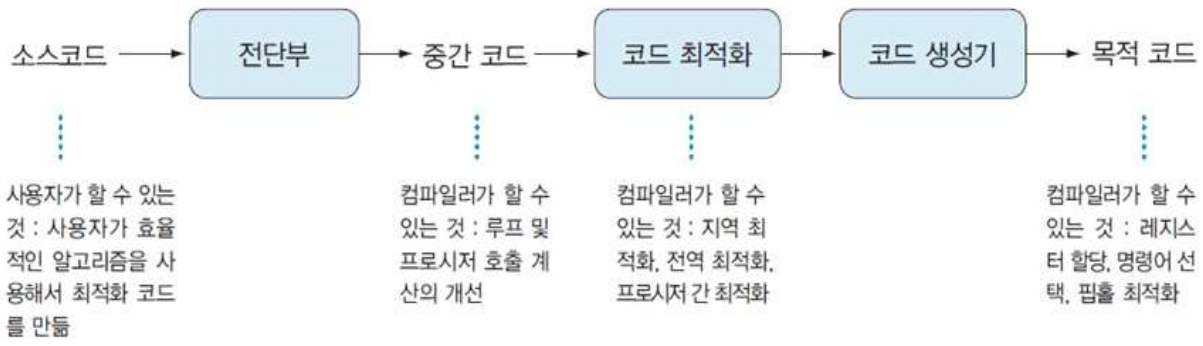


그림 13. 구현 단계에서의 적용되는 최적화 종류

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

코드 최적화 기법의 분류는 최적화 대상인 프로그램이 대상 하드웨어 시스템에 대한 의존성에 따른 기계 독립적 최적화와 기계 종속적 최적화로 크게 2가지로 나뉜다. 기계 독립적인 최적화 기법은 대상 프로그램의 범위에 따라 핍홀 최적화(Peephole Optimization), 지역 최적화, 전역 최적화, 내부 프로시저 최적화, 루프(반복문) 최적화 등이 있으며, 기계 종속적인 최적화는 시스템 특성에 따른 기법들이 있다. 핍홀 최적화는 몇 개의 연속적인 명령어를 하나의 명령어나 더 짧은 명령어로 변환하는 기법이며, 주로 목적 코드가 생성된 다음에 수행된다. 지역최적화는 부분적인 관점에서 일련의 비효율적인 코드를 구분해내고 좀 더 효율적인 코드로 만드는 기법이며, 최적화를 수행 영역인 기본 블록은 코드가 분기해 나가거나 분기해 들어오는 부분이 없는 부분으로 어떠한 제어 흐름도 가지고 있지 않으므로 분석이 거의 필요 없어 상대적으로 쉽게 최적화를 수행할 수 있다. 루프 최적화는 하나의 루프 안에서의 최적화 기법이다. 전체 코드의 10%가 실행 시간의 90%를 차지하는 부분이 루프이므로 루프 부분에 품질 좋은 코드를 생성하는 것은 매우 중요하다. 따라서 많은 횟수로 반복될 수 있는 루프는 가능한 계산복잡도가 낮은 내부 코드를 생성하는 것이 좋다. 전역 최적화는 기본 블록 범위를 벗어나지만 하나의 프로시저 내에서 비효율적인 코드를 찾아 좀 더 효율적인 코드로 만드는 기법이며, 이 기법은 지역 최적화보다 더 많은 정보와 비용이 필요해 상대적으로 수행하기가 더 어렵지만 효과는 훨씬 좋다. 이 전역 최적화는 자료의 흐름을 분석하여 프로그램 안에서 사용되는 변수의 적용 흐름에 관한 정보 수집을 통해서 이뤄진다. 프로시저 간 최적화는 전체 프로그림에 적용되는 모든 프로시저를 대상으로 프로시저간 관계에 적용되는 기법인데, 컴파일러가 관련 연계 정보 생성하면 링커가 이 정보를 이용해 최적화를 수행하는 방식이므로 링커 없이는 최적화가 전혀 이뤄지지 않는다. 따라서 매우 기본적인 프로시저 간 최적화만 수행하거나 아예 수행하지 않는다. 기계 종속적 최적화는 적용 대상의 하드웨어 시스템의 특성에 따라 매우 달라지는 기법으로 주로 메모리나 관련 명령어 처리 방법에 초점을 맞춘다. 이러한 분류의 최적화 기법에 적용 가능한 구체적인 최적화 기법은 표 23과 같다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

연번	대분류	중분류	소분류	중요도	용이성
1	기계 독립	핍홀	중복 명령어 제거		
2		핍홀, 전역	도달 불가능한 코드 제거		
3		핍홀	제어 흐름 최적화		
4		핍홀, 지역	대수학적 간소화		
5		핍홀, 루프	세기 감축		
6		핍홀	하드웨어 명령어 사용		
7		지역	공통 부분식 제거		쉬움
8		지역	복사 전파		쉬움
9		지역	죽은 코드 제거		쉬움
10		지역, 전역	상수 폴딩		쉬움
11		루프	코드 이동	매우 높음	
12		루프	세기 감축		
13		루프	귀납 변수 최적화		
14		루프	루프 융합		
15		루프	루프 교환		
16		루프	루프 전개		
17		전역	전역적 공통 부분식 제거	좋음	
18		프로시저	매개변수 전달 방법		매우 어려움
19		프로시저	비지역 변수의 참조		매우 어려움
20		프로시저	서로 호출 가능한 모든 프로시저 정보 확보		매우 어려움
21	기계 종속		중복된 LOAD 명령어 제거		
22			효율적인 명령어 선택		
23			레지스터 할당	매우 높음	어려움
24			레지스터 배정	매우 높음	어려움
25			명령어 스케줄링		

표 23. 최적화 기법 분류

[AMTC09]의 verification 클럭사이클인 60000 cycles를 1/10 수준인 6600cycles로 낮춘 [AJSP16]의 *F*함수를 기준으로 구현 코드를 살펴보면, 이러한 최적화 기법이 골고루 적용되어 있다. 특히 눈에 띄는 것은 도달 불가능한 코드 및 죽은 코드가 제거 되어있고,

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

연산 비용이 낮은 계산식이나 명령어가 적용될 수 있는 부분을 최대한 찾아 세기 감축이 적용되었으며, 공통 부분식을 최대한 제거하였고, 루프 부분에 대해 대대적인 개선이 이뤄져있으며, 128비트 기반의 레지스터에서 256비트 단위의 레지스터를 할당 및 배정하였음을 볼 수 있다. 이들 각각을 살펴보면 다음과 같다.

- 도달 불가능한 코드 및 죽은 코드 제거

  전체 코드에서 불필요한 코드가 포함될 수 있는데, 이 불필요한 코드는 도달 불가능한 코드와 죽은 코드를 일컫는다. 도달 불가능한 코드는 무조건 분기 바로 다음 문장에 있는 라벨이 없는 명령어 같은 것으로 도달할 수 없기 때문에 제거할 수 있다. 또는 어떤 변수가 특정 프로그램 지점 이후 전혀 사용하지 않는 값을 계산하는 문장이 있다면 이 부분도 불필요한 부분으로 제거될 수 있다. 예를 들어, 한 프로그램에 그림 14-15와 그림 16-17과 같은 코드들이 모두 포함 되어있는 경우에, 특정 수치로 한정지어 작성된 코드인 그림 14-15는 변수로 일반화되어 작성된 코드인 그림 16-17에 가려져 호출되지 않는다. 이 경우 그림 14-15와 같은 부분은 불필요한 코드 부분으로 제거할 수 있다. [AMTC09]에서 제공한 구현 코드에는 이러한 코드 부분이 다수 포함되어 있었으나, [AJSP16]에서는 이러한 불필요한 부분이 눈에 띄는 것이 거의 없다.

```
42  template <unsigned p>
43  class rainbow_w_56_40 {
44  public:
45      rainbow_w_56_40() {}
46      virtual ~rainbow_w_56_40(){}
47
48      static const unsigned l_msg = 56;
49      static const unsigned l_sec = 40;
50
51      static void pub_map( VEC<p,l_sec> *y , const uint8 * pub_key , const VEC<p,l_m
52      static size_t sizeof_pub_key() { return sizeof( MQPSv3<p, l_sec, l_msg> ); }
```

그림 14. 불필요한 코드 예1

```
71  size_t rainbow_w_56_40<p>::sizeof_q_key()
72  {
73      return  ( sizeof(MQPS< p , 16 , 16 >)
74                  + sizeof(MQPS< p , 8 , 32 >)
75                  + sizeof(MQPS< p , 16 , 40 >)
76                  + sizeof(MLPS< p , 16 , 16 >)
77                  + sizeof(MLPS< p , 8 , 8 >)
78                  + sizeof(MLPS< p , 16 , 16 >)
79                  + sizeof(MLPS< p , 16 , 16 >[16])
80                  + sizeof(MLPS< p , 8 , 32 >[8])
81                  + sizeof(MLPS< p , 16 , 40 >[16]) );
82  }
```

그림 15. 불필요한 코드 예2

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

```
676  template <unsigned p,unsigned V0,unsigned O0,unsigned O1>
677  class rainbow_w_N_M {
678  public:
679      rainbow_w_N_M() {}
680      virtual ~rainbow_w_N_M(){}
681
682      static const unsigned l_msg = V0+O0+O1;
683      static const unsigned l_sec = O0+O1;
684
685      static void pub_map( VEC<p,l_sec> *y , const uint8 * pub_key , const VEC<p,l_m
686      static size_t sizeof_pub_key() { return sizeof( MQPSv3<p, l_sec, l_msg> ); }
```

그림 16. 실행되는 코드 예1

```
704  template <unsigned p,unsigned V0,unsigned O0,unsigned O1>
705  size_t rainbow_w_N_M<p,V0,O0,O1>::sizeof_q_key()
706  {
707      return ( sizeof(MQPS< p , O0,V0 >)
708                  + sizeof(MLPS< p , O0,O0 >)
709                  + sizeof(MQPS< p , O1,V0+O0 >)
710                  + sizeof(MLPS< p , O1,O1 >)
711                  + sizeof(MLPS< p , O0,V0 >[O0])
712                  + sizeof(MLPS< p , O1,V0+O0 >[O1]) );
713
714  }
```

그림 17. 실행되는 코드 예2

- 세기 감축

세기 감축은 수행 비용이 높은 연산자를 수행 비용이 낮은 연산자로 대체하는 것이다. 예를 들어, 곱셈 연산시 피연산자가 $2n$이라면 곱셈 연산자를 쉬프트 연산자로 대체해 처리한다. 그림 18의 나머지 연산자를 이용한 모듈러 연산도 그림 19와 같이 비트와이즈 AND 연산자로 대체할 수 있다.

```
102                 if ((buf[j] % 31) != 31) {
103                     out[i] = (buf[j] % 31) - 15;
104                     i++;
105                 }
```

그림 18. 높은 비용의 모듈러 연산자 사용 예

```
112                 if ((buf[j] & 31) != 31) {
113                     out[i] = (buf[j] & 31) - 15;
114                     i++;
```

그림 19. 낮은 비용의 AND 연산자 사용 예

- 공통 부분식 제거

공통 부분식은 프로그램에서 공통된 부분이 여러 번 나타나는 경우에 해당 식이 한 번만 계산되고 나머지 부분은 제거함으로써 효율적인 코드를 만드는 것이다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

▪ 루프 개선

루프는 전체 코드의 10%가 실행 시간의 90%를 차지하는 부분이다. 이 루프를 개선하기 위하여 코드 이동, 세기 감축, 귀납 변수 최적화, 루프 융합/교환/전개 등을 수행할 수 있다. 그림 20의 내부 for의 종료 조건부분에 있는 계산식 또는 내부에 항상 같은 값을 가지고 있는 계산식을 루프 내부로 들어가기 전의 루프 외부로 이동시킬 수 있는데, 이러한 이동으로 계산 횟수를 대폭 줄일 수 있다.

```
957     for (unsigned k = 0; k < VEC<p, l_sec>::M; k++)
958         for (unsigned i = 0; i < (l_msg+1)*(l_msg+2)/2; i+=6){
959             const __m256i lo0 = _mm256_and_si256(_mm256_unpacklo_epi16(_polys[k][i
```

그림 20. 공통 부분식 포함 코드 예

```
957     MsgL = (l_msg+1)*(l_msg+2)/2;
958     __m256i mask = _mm256_set1_epi16(31);
959▼    for (unsigned k = 0; k < VEC<p, l_sec>::M; k++)
960▼        for (unsigned i = 0; i < MsgL; i+=6){
```

그림 21. 공통 부분식의 코드 이동 예

```
173     for (unsigned k = MQPSv3<p,m,n>::M; k--;){
174         __m128i acc = _mm_setzero_si128();
175
176         __m128i *q = (__m128i*)&poly->q[k][MQPSv3<p,m,n>::N];
177         for (unsigned i = N; i;){
178             __m128i wiji = wij[--i];
179             __m128i qqq = *--q;
180             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
181             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
182             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
183             qqq = *--q;
184             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
185             wiji = wij[--i];
186             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
187             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
188             qqq = *--q;
189             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
190             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
191             wiji = wij[--i];
192             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
193             qqq = *--q;
194             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
195             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
196             acc = _mm_add_epi32(acc, _mm_madd_epi16(_mm_shuffle_epi32(wiji,
197         }
198         acc = _mm_add_epi32(_mm_srai_epi32(acc, 10), _mm_and_si128(acc, mas
199         if (k & 1)
200             acchi = acc;
201         else
202             *--res = reducev((gfv<p>)_mm_packs_epi32(acc, acchi));
203     }
```

그림 22. 중첩 루프 부분 예

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

또한, 프로그램의 고속화를 위해서는 어느 정도의 반복은 루프 없이 코드를 직접 차례로 작성하는 것이 효과적이다. [AMTC09]의 구현 코드 경우에는 그림 20과 같이 중첩 for의 사용이 많으나, [AJSP16]의 경우에는 중첩 루프가 많이 배제되고, 루프의 계산을 빠르게 하기 위해 루프를 펼치는 효과를 내는 방법으로 루프 전개가 적용되었으며, 경우에 따라서는 루프를 해제하여 루프없이 전개하는 코드로 프로그램 수행 속도의 향상을 도모하였다.

- 레지스터 할당 및 배정

본 연구에서 진행하는 다변수 이차식 기반의 암호 및 서명 알고리즘을 효과적으로 구현하기 위해서 대부분의 연구자는 레지스터를 사용하고 있다. 레지스터는 기계에 강하게 종속적인 특성을 갖는데, CPU 칩의 발달로 사용 가능한 레지스터가 64비트 -> 128비트 -> 256비트 ->512비트로 발달하고 있으며, 이와 관련하여 구현이 용이하도록 각 칩 제조사는 전용 벡터 확장 명령 집합을 구성하여 내주고 있다. 현재 보편적으로 사용할 수 있는 레지스터는 128비트 또는 256비트로 그림 22의 경우에는 128비트 레지스터를 사용하는 _mm_로 시작하는 명령어가 적용되어 구현되었으며, 그림 23은 256비트 레지스터를 사용하는 _mm256_로 시작하는 명령어를 이용해 구현하여 프로그램 수행 속도를 향상시켰다.

```
309     r_x1 = _mm256_maddubs_epi16(y1, x1);
310     r_x2 = _mm256_maddubs_epi16(y2, x2);
311     r_x1 = reduce_16(r_x1, mask_reduce, mask_2114);
312     r_x2 = reduce_16(r_x2, mask_reduce, mask_2114);
313     _mm256_store_si256((__m256i*)cc, _mm256_packs_epi16(r_x1, r_x2));
314     cc += 32;
315
316     __m256i upside_down_x1 = _mm256_permute4x64_epi64(x1, 0x4e);
317     __m256i upside_down_x2 = _mm256_permute4x64_epi64(x2, 0x4e);
318
319     r_x1 = _mm256_maddubs_epi16(y1, _mm256_alignr_epi8(upside_down_x1, x1, 2));
320     r_x2 = _mm256_maddubs_epi16(y2, _mm256_alignr_epi8(upside_down_x2, x2, 2));
321     r_x1 = reduce_16(r_x1, mask_reduce, mask_2114);
322     r_x2 = reduce_16(r_x2, mask_reduce, mask_2114);
323     _mm256_store_si256((__m256i*)cc, _mm256_packs_epi16(r_x1, r_x2));
324     cc += 32;
325
326     r_x1 = _mm256_maddubs_epi16(y1, _mm256_alignr_epi8(upside_down_x1, x1, 4));
327     r_x2 = _mm256_maddubs_epi16(y2, _mm256_alignr_epi8(upside_down_x2, x2, 4));
328     r_x1 = reduce_16(r_x1, mask_reduce, mask_2114);
329     r_x2 = reduce_16(r_x2, mask_reduce, mask_2114);
330     _mm256_store_si256((__m256i*)cc, _mm256_packs_epi16(r_x1, r_x2));
331     cc += 32;
332
333     r_x1 = _mm256_maddubs_epi16(y1, _mm256_alignr_epi8(upside_down_x1, x1, 6));
334     r_x2 = _mm256_maddubs_epi16(y2, _mm256_alignr_epi8(upside_down_x2, x2, 6));
```

그림 23. 256비트 레지스터 적용 예

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

3. SIMD 병렬 처리 기반의 고속 프로그래밍

가. 병렬 처리 기반 프로그래밍 개요

다변수 이차식 기반의 알고리즘에 대한 고속화를 실현하기 위한 대표적인 방법은 한 번의 연산으로 여러 개의 데이터를 처리하는 SIMD(Single Instruction Multiple Data) 기반으로 프로그래밍하는 것이다. 대부분의 CPU는 이 SIMD를 제공하고 있으나, 수많은 사람들은 통상적으로 한 번에 한 개의 데이터만 처리할 수 있는 SISD(Single Instruction Single Data) 기반의 구조로 프로그래밍 하고 있다. 더 나아가 MIMD(Multiple Instruction Multiple Data)는 여러 개의 연산 명령으로 여러 개의 데이터를 처리한다. 이러한 병렬처리 모델 SIMD 또는 MIMD 기반의 프로그램 구현이 가능하도록 하는 현재의 시스템 구성은 하나의 CPU에 연산을 처리하는 코어가 8개, 레지스터 메모리가 256bit씩 16개, 크기가 더 커진 캐쉬 등이 올려져있으며(멀티코어 CPU), 이러한 CPU가 다중으로 탑재기도 하고(슈퍼컴퓨터 또는 워크스테이션), 그래픽 처리 능력을 향상시키기 위해 수백 개의 GPU를 넣은 그래픽카드를 제작해 탑재하기도 한다(멀티코어 GPU).

한 연산에 여러 데이터를 처리하거나 여러 연산에 여러 데이터를 처리하는 병렬 프로그램 모델은 명령어 수준 병렬 프로그래밍 모델, 공유 메모리 병렬 프로그래밍 모델, 메시지 패싱 병렬 프로그래밍 모델, 그리고 이 모델들을 융합한 하이브리드 모델로 크게 4가지이다. 명령어 수준 병렬 프로그래밍 모델은 SIMD로 대표되며 작은 수준의 병렬 프로그래밍이다. AMD와 Intel의 대부분의 CPU는 병렬처리 SIMD를 지원하며, 이러한 CPU를 탑재한 대다수의 PC는 컴퓨터 하드웨어의 특성을 적게 받아 가장 간편하게 병렬처리 기능을 활용해 볼 수 있다. 공유 메모리 병렬 프로그래밍 모델은 연산 작업을 처리하는 CPU가 여러 개 있어 다중 CPU 기반의 멀티스레드로 데이터를 병렬처리한다. 또한 다수의 GPU가 있는 그래픽 카드를 장착하여 병렬처리의 효과를 얻을 수 있다. 이 모델에는 멀티스레딩, pthreads, OpenMP, CUDA 등이 해당된다. 메시지 패싱 병렬 프로그래밍 모델은 여러 개의 PC와 이 PC들 간의 통신으로 대용량 데이터를 처리하는 것으로 분산 컴퓨팅 또는 그리드 컴퓨팅이라고도 불리며, MPI, HPF, PVM 등이 해당된다. 하이브리드 모델은 앞의 세 가지 모델을 혼합하여 사용한 것으로 더욱 큰 효과를 얻을 수 있으며, OpenMP+MMPI, SIMD+멀티스레딩, GPU+CPU 등의 모델이 해당된다.

이러한 병렬 프로그래밍 모델 중에서 암호 알고리즘의 고속화 구현을 위해 주로 적용되고 있는 방식은 SIMD이다. 특히, 동일한 연산이 반복적으로 수행되는 부분이 많은 대칭키 기반의 암호 알고리즘에서 주로 속도향상을 위해 이 SIMD를 적용하였다. 반면에 반복 보다는 수학 모델 기반의 연산이 적용된 공개키 암호 알고리즘에서는 SIMD를 크게 고려하지 않았다. 그러나, Post Quantum 암호 알고리즘의 하나로 최근에 이슈가 되고 있는 다변수 이차식 기반 알고리즘은 행렬 기반의 반복적인 연산이 주를 이루고 있어 SIMD를 적용해 제안 알고리즘의 속도를 높이는데 크게 기여한다. 최근에 나오는 논문일수록 최근의 시스템을 사용하여 향상된 SIMD 명령어의 사용 및 효과적인 구현 결과를

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

보이고 있다[AJSP16].

나. SIMD 개요

SIMD 기반의 연산은 그림 24와 같이 하나의 변수에 여러 개의 데이터를 배정하고, 한 번의 연산 명령으로 해당 데이터들의 연산이 모두 수행이 되는 방식이다. 즉, 이 방식의 기존 방식과의 차이는 한 번의 CPU 클럭 동작으로 하나의 결과와 여러 개의 결과가 나오는 차이를 의미한다. SIMD 기반에서 사용되는 변수는 레지스터를 사용하며, 이 레지스터는 CPU의 발전에 따라 그 크기가 64비트-> 128비트-> 256비트-> 512비트로 확대 되었고, 이와 관련한 벡터 확장 명령 집합을 MMX, SSE, AVX2, AVX3(512) 등으로 제조사에서 제공하고 있다. 현재 주로 사용 가능한 명령집합은 SSE와 AVX2이다. 128비트 크기의 레지스터 XMM의 개수는 32bit 시스템에서는 총 8개로 XMM0, XMM1~XMM7을 사용할 수 있으며, 64bit 시스템에서는 XMM8~XMM15로 16개가 증가되었으므로 프로그램 구현시 최대한 사용할수록 효율이 높아진다. 최근에는 이 레지스터가 256비트 및 512비트 기반의 레지스터로 확장되었으며, 그에 따라 256비트 레지스터를 기반으로 하는 SIMD 프로그램 구현 시도에 따른 결과가 나오기 시작하고 있으나, 512비트 기반 레지스터의 경우에는 적용된 시스템이 한정적이어서 다변수 이차식 기반 알고리즘에 적용된 실험 결과가 발표된 것은 없다.

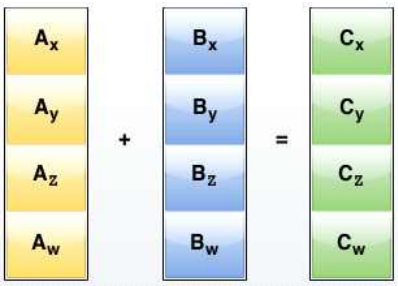


그림 24. SIMD 기반 연산

일반적으로 코딩해오던 SISD 방식과 SIMD의 성능 차이를 살펴보면, 다음과 같다. 보통 연산 속도는 병렬처리되는 패킹 사이즈에 영향을 받게 되는데, 여기서는 단일 연산과 비교한다. 32비트 int형 정수는 128비트 레지스터에 4개가 들어갈 수 있어 한번에 4개의 연산을 동시에 수행하게 되고, 이때 단일 int형 연산시보다 10~30%가 연산속도가 빨라진다. 이 128비트 레지스터에 들어갈 수 있는 pack의 개수(short면 8개, byte면 16개)에 따라 성능의 차이가 비약적으로 증가(short면 2배, byte면 4~6배)한다. 2016년에 발표된 연구 결과 [AJSP16]에서는 256비트 레지스터 사용으로 short형은 16개, byte형은 32개를 패킹할 수 있었고, 그에 따라 향상된 성능을 확보할 수 있었다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Packing 사이즈	byte	short	int
병렬 연산 개수	16	8	4
C 코드와 성능 차이	4~6배	2배	10~30%

표 24. SISD와 SIMD의 성능차이 (단일 연산 기준)

SIMD 프로그램 순서는 128비트 기반으로 살펴보면 그림 25와 같다.

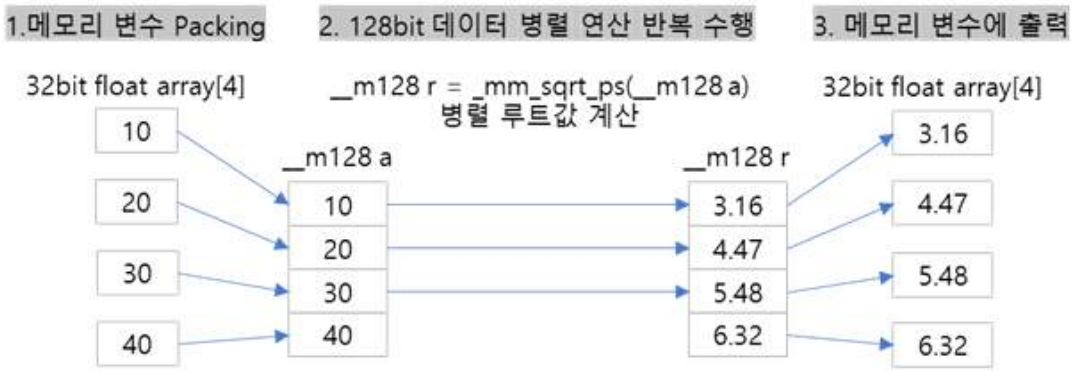


그림 25. SIMD 프로그램 순서

SIMD 구현 방법은 어셈블리 언어로 SIMD 명령어를 이용하는 방법, C 언어로 intrinsic 함수를 이용하는 방법, C++ 언어로 Vector Class를 이용하는 방법으로 3가지가 있다. 이 중에서 프로그램 구현의 편의성과 성능 향상성을 살펴보면 그림 19와 같다. SIMD 구현은 일반적인 C/C++ 프로그램 대비 편의성은 낮은 편이지만, 세 가지 방법을 상대적으로 살펴보면 어셈블리 언어를 이용하는 것이 편이성이 가장 낮다. 하지만 어셈블리 언어를 이용하여 SIMD를 구현하면 세세한 부분도 제어 가능해 상대적으로 구현 편이성이 높은 Vector Class보다 안정적인 성능을 보인다. Intrinsic 함수는 어셈블리 SIMD 명령어를 사용하기 좋도록 C의 함수로 만들어놓은 것이다. 따라서, 다른 C 함수 사용과 같아 빠르게 숙지하고 구현할 수 있으나 레지스터에 값을 넣는 방법과 계산하는 순서는 컴파일러가 관여한다. 즉, 컴파일러가 사용된 Intrinsic 함수를 기준으로 어떤 XMM 레지스터(128비트 레지스터)에 어떤 값을 넣고 어떤 순서로 계산할지를 결정한다. Intrinsic 함수 이용에서 팩(Pack) 방식의 연산이 수행되는 함수와 스칼라 방식의 연산이 수행되는 함수가 있는데, 팩 연산의 함수는 어셈블리 언어와 동등한 속도를 발휘하지만, 스칼라 연산의 함수는 어셈블리 언어보다 약간 느린 특성이 있다. 따라서, 팩 중심의 프로그램 구현인 경우는 크게 차이나지 않는다.

- 팩 연산 방식
  여러 개의 팩(여러 개의 데이터)을 한 번의 명령만으로 연산
- 스칼라 연산 방식
  최하위 팩만 연산하고 나머지 팩은 앞선 인자의 값을 그대로 전달

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

Vector Class는 intrinsic 함수를 wrapping하여 만든 클래스로 연산자 오버로딩 등을 지원하므로 상대적으로 사용하기 편리하지만, 객체화하면서 생성되는 오버헤드(상속, 오버로딩, 가상함수, 임시 객체 생성 등)가 높아 효율성이 상대적으로 떨어진다.

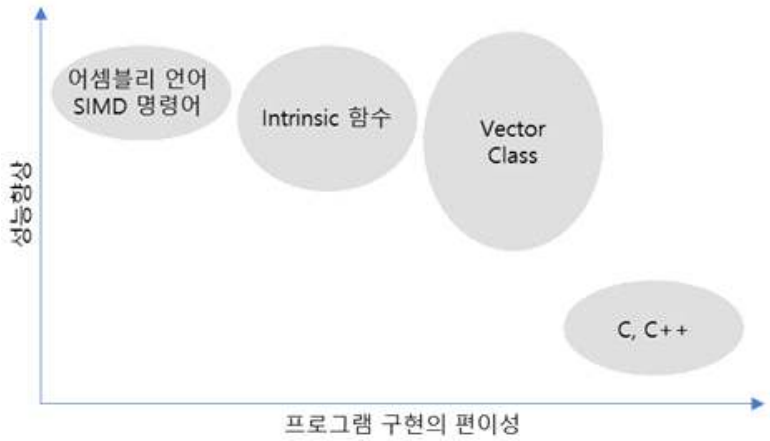


그림 26. SIMD 구현 방법에 따른 편이성 vs 성능향상

SIMD를 프로그램에 적용하기 위해서는 프로그램의 핵심 부분 및 병목 지점, 프로그램 구조의 SIMD 적합성, 성능 향상의 예측치 계산, 자료구조(정수형, 실수형) 구분, 128bit에 저장 가능한 데이터 개수, 프로그램 제작/디버깅/테스팅 기간 고려, 구현 도구 결정, SISD와의 성능 비교 테스트 등의 과정이 필요하며 고속화를 효과적으로 구현 가능하다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 4 장. 연구개발 목표 달성도 및 대외 기여도

<table>
<tr><th>목표
(달성도)</th><th>내용</th></tr>
<tr><td>다양한 난제 기반 공개키 암호알고리즘 비교 분석 (100%)</td><td>

○다양한 난제 기반 공개키 암호알고리즘 경량기기에서 요구조건/문제점/효율성 비교분석 및 도전과제 도출

-리소스가 제한된 8-bit, 16-bit, 32-bit microprocessor에서 다양한 난제 기반 공개키암호 설계방법, 요구조건, 문제점, 효율성 비교 분석 및 도전과제 도출

<table>
<tr><th></th><th>상세</th><th>전자서명</th><th>서명생성</th><th>서명 검증</th><th>기반 문제</th></tr>
<tr><td rowspan="5">8-bit</td><td>AX 128</td><td>Rainbow(36,21,22)</td><td>8,227,000</td><td>9,216,000</td><td>MQ 문제,
IP 문제,
MinRank 문제</td></tr>
<tr><td>AX 128</td><td>enTTS(15,60,88)</td><td>2,142,000</td><td>30,780,000</td><td>MQ 문제,
IP 문제,
MinRank 문제</td></tr>
<tr><td>AX 128</td><td>BLISS-I</td><td>10,537,981</td><td>2,814,118</td><td>Ring-SIS 문제</td></tr>
<tr><td>AT 2560</td><td>Curve 25519</td><td>13,900,397</td><td></td><td>ECDLP</td></tr>
<tr><td>AT 2560</td><td>NIST P-256</td><td>17,520,000</td><td></td><td>ECDLP</td></tr>
<tr><td rowspan="4">16-bit</td><td rowspan="2">MSP 430X 16-bit H/W multiplier</td><td>NIST P-256</td><td>7,284,377</td><td></td><td>ECDLP</td></tr>
<tr><td>Curve 25519</td><td>7,933,296</td><td></td><td>ECDLP</td></tr>
<tr><td rowspan="2">MSP 430X 16-bit H/W multiplier</td><td>NIST P-256</td><td>5,321,776</td><td></td><td>ECDLP</td></tr>
<tr><td>Curve 25519</td><td>5,301,792</td><td></td><td>ECDLP</td></tr>
<tr><td rowspan="4">32-bit</td><td rowspan="2">ARM Cortex M3</td><td>SPHINCS 256</td><td>729,942,616</td><td>17,707,814</td><td>hash collision resistant, 2nd preimage re-sistant</td></tr>
<tr><td>XMSS^{MT}</td><td>22,725,092</td><td>5,528,712</td><td>hash collision</td></tr>
<tr><td rowspan="2">ARM Cortex M0</td><td>NIST P-256</td><td>2,762,000</td><td></td><td>ECDLP</td></tr>
<tr><td>Curve 25519</td><td>3,589,850</td><td></td><td>ECDLP</td></tr>
</table>

표 25. 다양한 난제 기반 공개키 암호 효율성 비교 분석

※ 위 표에 있는 전자서명 알고리즘의 상세는 다음과 같음.

- NIST P-256: NIST 표준 타원곡선, Curve25519: Twisted Edwards curve
- SPHICS 256: 해쉬함수 기반 stateless 전자서명 알고리즘
- XMSS^{MT}: 해쉬함수 기반 stateful 전자서명 알고리즘
- BLISS-I: 격자기반 전자서명
- Rainbow: 다변수 이차식 기반 전자서명,
- enTTS: 3-layer Rainbow의 특별한 경우
- ECDLP: 타원곡선 이산대수 문제
- AX 128: ATxmega 128A1, 32MHz
- AT 2560: ATmega 2560, 16 MHz

</td></tr>
</table>

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

목표 (달성도)	내용
다변수 이차식 기반 암호알고리즘 안전성 분석(100%)	○ **다변수 이차식 전자서명 RGB 키 복구 공격에 대한 안전성 분석** -Shen과 Tang의 Computer Journal에 게재된 논문 "RGB, a Mixed Multivariate Signature Scheme"과 Tang, Lv, Shen이 국제 암호학회 ACISP 2016에서 발표한 "Hybrid MQ Signature for Embedded Device"에서 제안된 다변수 이차식 기반 전자서명 RGB 안전성 분석 -RGB의 affine map T는 (r,g,b)의 변수를 모두 섞어주지 않고 T_1은 r개의 원소만, T_2는 나머지 (g,b)의 원소만 섞어주고 있다는 성질과 F의 계수가 0인 부분을 이용한 equivalent key와 good key를 이용한 키 복구 공격에 안전하지 않다는 것을 보임. -**제안된 RGB 모든 파라미터는 수 시간 안에 모두 깨지**는 것을 볼 수 있음.

(r,g,b)	안전도 (λ)	good key를 이용한 KRA 복잡도	good key를 이용한 KRA 공격시간(초)
(16, 18, 10)	64	2^{23}	0.48
(24, 24, 12)	80	2^{25}	1.7
(30, 32, 16)	96	2^{32}	90.68
(32, 34, 18)	118	2^{34}	660.37
(40, 35, 20)	128	2^{39}	24551.16

표 26. RGB 파라미터에 대한 키 복구 공격에 대한 복잡도와 실제 공격시간

○ **다변수 이차식 전자서명 RGB의 UOV 공격에 대한 안전성 분석**

-UOV 공격에 안전하기 위해서는 $b \geq g + \frac{\lambda}{8}$를 만족해야 함을 보임. 위의 파라미터는 모두 $b < g$이므로 UOV 공격에 안전하지 않음.

○ **RGB의 공격에 안전한 파라미터 선택**

-키 복구 공격과 다른 모든 공격에 안전한 파라미터 선택

(r,g,b)	안전도	키 복구 공격	Direct 공격	UOV 공격
(16, 24, 31)	64	2^{130}	2^{69}	2^{66}
(24, 29, 38)	80	2^{161}	2^{80}	2^{83}
(30, 35, 46)	96	2^{191}	2^{96}	2^{100}
(32, 43, 57)	118	2^{237}	2^{119}	2^{125}
(40, 47, 62)	128	2^{253}	2^{130}	2^{134}

표 27. RGB의 공격에 안전한 파라미터

-안전한 파라미터에서의 RGB의 서명생성과 검증 속도 외 키 길이가 UOV와 Rainbow 보다 2-3배 더 느리고, 길어, 효율성이 떨어짐.

안전도	전자서명	서명길이(B)	공개키(kB)	비밀키(kB)
128	UOV	103	234.6	201.5
	Rainbow	79	136.1	102.5
	RGB	109	449.7	422.496

표 28. RGB, UOV, Rainbow 서명길이, 키 길이 비교

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

<table>
<tr><th>목표
(달성도)</th><th>내용</th></tr>
<tr><td></td><td>
<table>
<tr><th>안전도</th><th>전자서명</th><th>서명 생성(cycle)</th><th>서명 검증(cycle)</th></tr>
<tr><td rowspan="3">128</td><td>UOV</td><td>237,395</td><td>244,728</td></tr>
<tr><td>Rainbow</td><td>113,383</td><td>142,437</td></tr>
<tr><td>RGB</td><td>469,204</td><td>466,900</td></tr>
</table>
표 29. RGB, UOV, Rainbow에서 요구되는 곱셈의 수 비교
</td></tr>
<tr><td>다변수 이차식 기반 고속 전자서명 알고리즘 설계 및 안전성 분석(100%)</td><td>
○ <u>키 길이를 최소화 시킨 새로운 다변수 이차식 기반 고속 전자서명 알고리즘 HiMQ 설계</u>
-공개키의 크기를 기존 다변수 이차식 기반 전자서명보다 늘이지 않으면서 비밀키를 최소화 시킬 수 있는 central map 설계
-비밀키 길이의 최소화: 알려진 rank 공격에 안전한 rank를 가지는데 필요한 충분한 이차항을 가지면서, 이차식 계산과 키 길이를 최소화하기 위한 sparse한 다항식으로 구성된 2-layer 구조
-서명생성 복잡도 최소화: linear system의 해를 찾는 Gaussian elimination의 복잡도가 $O(n^3)$으로 가장 시간이 걸리는 부분이므로, 해를 쉽게 구할 수 있는 이차식의 시스템을 이용하여 Gaussian elimination의 횟수를 단 한번으로 줄일 수 있는 central map 설계
-HiMQ의 공개키 길이는 F_{2^8}의 $\frac{m(n+1)(n+2)}{2}$ 개의 원소로, 비밀키는 $o_1(v_1+1)+2o_2(v_1+o_1)+n(v_1+o_1+1)+m(m+1)-o_2^2$개의 원소로 이루어짐..
○ <u>설계한 다변수 이차식 기반 전자서명 HiMQ 안전성 분석</u>
-키 복구 공격에 대한 안전성 분석: good key를 이용한 키 복구 공격의 복잡도가 $m+n-1$개의 식과 n개의 변수로 이루어진 다변수 이차식 시스템의 해를 구하는 복잡도와 동치인 것을 보임.
-Direct 공격에 대한 안전성 분석; MQ-문제를 풀어주는 가장 효율적인 알고리즘 HybridF5에 안전함을 보임.
-UOV 공격에 대한 안전성 분석: central map 를 구성하는 이차식을 대칭행렬로 표현했을 때 그것들의 선형 결합을 이용한 invariant subspace를 찾을 복잡도가 $q^{v+o_1-o_2}$임을 보임.
-MinRank 공격에 대한 안전성 분석: 첫번째 layer를 구성하는 대칭행렬의 선형결합의 rank가 최소가 되는 선형결합을 찾을 복잡도가 $o_1 \cdot q^{v-o_1+3}$ 임을 보임.
-HighRank 공격에 대한 안전성 분석: 마지막 layer를 구성하는 대칭행렬의 선형결합의 rank가 full rank가 나오지 않을 복잡도가 $q^{o_2} \cdot \frac{n^3}{6}$임을 보임.
</td></tr>
</table>

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 5 장. 연구개발 결과의 활용계획

- 기존 공개키 암호알고리즘의 안전성 분석 및 취약성 분석 결과는 새로 제안하는 공개키 암호알고리즘의 설계 기준으로 활용될 것이며, 설계된 고속 전자서명은 초소형, 저전력 등을 요구하는 스마트 기기 인증에 활용할 수 있음.

- 다변수 이차식으로 구성된 시스템의 해를 구하는 문제에 기반한 고속 공개키 암호알고리즘은 현재 국제표준으로 제정되어 있는 공개키 암호 RSA와 ECC가 너무 느려서 사용하지 못했던 경량기기에 사용 가능한 후보가 될 것임.

- 신규 수학적 문제를 이용한 고속 공개키 암호알고리즘의 개발을 통해 경쟁력 있는 기술 확보가 가능하고, 개발된 고속 공개키 암호는 다양한 응용환경에서 요구하는 특수한 목적과 기능을 수행할 수 있는 암호 응용기술에 기반이 되는 원천기술로 활용될 것임.

- 사업의 결과로 나온 암호알고리즘은 국제 표준화 기구 ISO/IEC에서 향후 표준화 후보 item으로 논의되고 있는 것 중 long-term security를 제공하는 양자 컴퓨터에 안전한 암호알고리즘을 겨냥한 “Achieving long-term security with unconditional schemes” 분야의 국제 표준화 추진이 가능하며, 암호알고리즘 구현코드는 오픈소스 라이브러리 형태로 공개하여 학술적 연구와 산업체에서의 실질적인 활용이 가능하도록 할 예정임.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 6 장. 연구과정에서 수집한 해외 과학기술 정보

※ 연구과정에서 수입한 해외 과학 기술 정보는 제 2 장 국내 기술 개발 현황에 모두 포함되어 있음.

제 7 장. 국가 과학기술종합정보시스템에 등록한 연구시설·장비 현황

구입 기관	연구시설/ 연구장비명	규격 (모델명)	수량	구입 연월일	구입 가격 (천원)	구입처 (전화번호)	비고 (설치 장소)	NTIS장비 등록 번호
NIMS	암호알고리즘 분석용 서버	PowerEdge R730xd	1	2016.10.19	38,700	㈜다솜코리아 (031-766-7618)	연구소 전산실	NFEC-2016-10-212540

제 8 장. 연구개발과제의 대표적 연구 실적

번호	구분 (논문/ 특허/ 기타)	논문명/특허명/ 기타	소속 기관명	역할	논문 게재지/ 특허 등록 국가	영향력 지수	논문 게재일 /특허 등록일	사사 여부 (단독 사사 또는 중복 사사)	특기 사항 (SCI 여부/인용 횟수 등)
1	논문	A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks	NIMS	주 저자	IEEE Communications Surveys and Tutorials	IF 9.220	2016	단독 사사	SCI/7회 인용
2	논문	On r-th Root Extraction Algorithm in F_q for $q \equiv lr^s + 1 \mod r^{s+1}$ with $1 < l < r$ and Small s	NIMS	주 저자	IEEE Trans. on Computers	IF 1.723	2016	중복 사사	SCI
3	논문	Cryptanalysis of RGB, a Mixed Multivariate Quadratic Signature Scheme	NIMS	주 저자	Finite Fields and Their Applications	IF 1.292	게재 예정	단독 사사	SCI

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

제 9 장. 참고문헌

[ALSK15] R. Azarderakhsh, Z. Liu, H. Seo, and Howon Kim, NEON PQCryto: Fast and Parallel Ring-LWE Encryption on ARM NEON Architecture, Cryptology ePrint Archive: Report 2015/1081, 2015.

[AMTC09] A. I. Chen, M. Chen, T. Chen, C. Cheng, J. Ding, E. L. Kuo, F. Y. Lee, and B. Y. Yang, SSE implementation of multivariate PKCs on modern x86 CPUs, CHES 2009. Springer Berlin Heidelberg, pp. 33-48, 2009.

[BBDM10] J. Buchmann, S. Bulygin, J. Ding, W. S. A. E. Mohamed, and F. Werner, "Practical Algebraic Cryptanalysis for Dragon-Based Cryptosystems", CANS 2010, LNCS 6467, pp. 140-155, 2010.

[BCP97] W. Bosma, J. Cannon, and C. Playoust, The Magma algebra system. I. The user language, J. Symbolic Comput., 24 (1997), 235 - 265.

[BFP08] L. Bettale, J.-C. Faugère, and L. Perret, Hybrid Approach for Solving Multivariate Systems over Finite Fields, Journal of Mathematical Cryptology. 2, pp. 1-22, 2008.

[BFS97] J. F. Buss, G. S. Frandsen, and J. O. Shallit, The Computational Complexity of Some Problems of Linear Algebra, STACS 1997, LNCS 1200, pp. 451-462, Springer, 1997.

[BG06] O. Billet and H. Gilbert, Cryptanalysis of Rainbow. SCN 2006, LNCS 4116, pp. 336-347. Springer, 2006.

[CHRS16] M. S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska and P. Schwabe, From 5-pass MQ-based identification to MQ-based signatures, ASIACRYPT 2016, LNCS 10032, pp. 135-165, Springer, 2016.

[CHT12] Peter Czypek, Stefan Heyse, and Enrico Thomae, Efficient Implementations of MQPKS on Constrained Devices, CHES 2012, LNCS 7428, pp. 374 - 389, 2012.

[CSV94] D. Coppersmith, J. Stern and S. Vaudenay, Attacks on the birational signature scheme. CRYPTO 1993, LNCS 773, pp. 435-443. Springer, 1994.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, Lattice Signatures and Bimodal Gaussians, CRYPTO 2013, Part I, LNCS 8042, pp. 40-56, 2013.

[DHHH15] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe, High-speed Curve25519 on 8-bit, 16-bit and 32-bit microcontrollers, Designs, Codes and Cryptography, 77(2-3), pp. 493-514.

[DPWT16] D. H. Duong, A. Petzoldt, Y. Wang, and T. Takagi, Revisiting the Cubic UOV Signature Scheme, IACR eprint archive, http://eprint.iacr.org/2016/1079, 2016.

[DS05] J. Ding and D. Schmidt, Rainbow, a New Multivariable Polynomial Signature Scheme, ACNS 2005, LNCS 3531, pp. 164-175, Springer, 2005.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

[DY13] J. Ding, and B. -Y. Yang, "Degree of Regularity for HFEv and HFEv-, PQCrypto 2013, LNCS 7932, pp.52-66, 2013.
[DYCC08] J. Ding, B. -Y. Yang, C. -H. O. Chen, M. -S. Chen, and C. -M. Cheng, New Differential Algebraic Attacks and Reparametrization of Rainbow, ACNS 2008, LNCS 5037, pp. 242-257, 2008.
[FGPS15] J. -C. Faugere, D. Gligoroski, L. Perret, S. Samardjiska, and E. Thomae, A Polynomial-Time Key-Recovery Attack on MQQ Cryptosystems, PKC 2015, LNCS 9020, pp. 150-174, 2015
[FJ03] J. -C. Faugere, and A. Joux, Algebraic Cryptanalysis of Hidden Field Equation(HFE) Cryptosystems Using Gröbner Bases, Crypto 2003, LNCS 2729, pp. 44-60, 2003.
[G96] L. K. Grover, A fast quantum mechanical algorithm for database search, STOC 1996, pp. 212 - 219, ACM, 1996.
[GC00] L. Goubin and N. T. Courtois, Cryptanalysis of the TTM Cryptosystem, ASIACRYPT 2000, LNCS 1976, pp. 44-57, Springer, 2000.
[GMK08] D. Gligoroski, S. Markovski, and S. J. Knapskog, A Public Key Block Cipher Based on Multivariate Quadratic Quasigroups, Cryptology ePrint Archive 2008/210.
[GOJP12] D. Gligoroski, R. S. Odegard, R. E. Jensen, L. Perret, J. -C. Faugere, S. J. Knapskog, and S. Markovski, MQQ-SIG, An Ultra-Fast and Provably CMA Resistant Digital Signature Scheme, INTRUST 2011, LNCS 7222, pp. 184-203, 2012.
[GS12] D. Gligoroski, and S. Samardjiska, The Multivariate Probabilistic Encryption Scheme MQQ-ENC, Cryptology ePrint Archive 2012/328.
[GOL12] C. P. L. Gouvêa, L. B. Oliveira, and J. López. Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller, Journal of Cryptographic Engineering, 2(1), 2012.
[Gu16] C. Gu, Cryptanalysis of Simple Matrix Scheme for Encryption, Cryptology ePrint Archive 2016/1075.
[H16] Y. Hashimoto, On the security of Cubic UOV. IACR eprint archive, http://eprint.iacr.org/2016/788, 2016.
[Has16] Y. Hashimoto, A note on Tensor Simple Matrix Encryption Scheme, Cryptology ePrint Archive 2016/065.
[HRSS16] A. Hülsing, J. Rijneveld, S. Samardjiska, P. Schwabe, From 5-pass MQ-based identification to MQ-based signatures, IACR Cryptology ePrint Archive, Report 2016/708, 2016.
[KPG99] A. Kipnis, J. Patarin, and A. Shamir, Unbalanced Oil and Vinegar Signature Scheme, Eurocrypt 1999, LNCS 1592, pp. 206-222, 1999.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

[KS98] A. Kipnis, and A. Shamir, Cryptanalysis of the Oil and Vinegar Signature Scheme, Crypto 1998, LNCS 1462, pp. 257-266, 1998.
[KS99] A. Kipnis, and A. Shamir, Cryptanalysis of the HFE Public Key Cryptosystem, Crypto 1999, LNCS 1666, pp. 19-30, 1999.
[LSRG15] Z. Liu, H. Seo, S. S. Roy, J. Großschädl, H. Kim, and I. Verbauwhede, Efficient Ring-LWE Encryption on 8-Bit AVR Processors, CHES 2015, LNCS 9293, pp 663-682, 2015.
[MDBW09] M. S. E. Mohamed, J. Ding, J. Buchmann, and F. Werner, Algebraic Attack on the MQQ Public Key Cryptosystem, CANS 2009, LNCS 5888, pp. 392-401, 2009.
[NLXL15] X. Nie, B. Liu, H. Xiong, G. Lu: Cubic unbalance oil and vinegar signature scheme. Inscrypt 2015, LNCS 9589, pp. 47-56. Springer, 2016.
[P96] J. Patarin, Hidden Fields Equations (HFE) and Isomorphism of Polynomials (IP) : Two New Families of Asymmetric Algorithms, Eurocrypt 1996, LNCS 1070, pp. 33-48, 1996.
[P97] J. Patarin, The Oil and Vinegar Algorithm for Signatures, presented at the Dagstuhl Workshop on Cryptography, September 1997.
[PCG01] J. Patarin, N. Courtois, and L. Goubin, QUARTZ, 128-Bit Log Digital Signatures, CT-RSA 2001, LNCS 2020, pp. 282-297, 2001.
[PTCW16] Z. Peng, S. Tang, J. Chen, C. Wu, and X. Zhang, Fast Implementation of Simple Matrix Encryption Scheme on Modern x64 CPU, ISPEC 2016, LNCS 10060, pp.151-166, 2016.
[PCYT15] A. Petzoldt, M. -S. Chen, B. -Y. Yang, C. Tao, and J. Ding, Design Principles for HFEv- based Multivariate Signature Schemes, Asiacrypt 2015 Part I, LNCS 9452, pp. 311-334, 2015.
[PDW16] A. Petzoldt, J. Ding, and L. -C. Wang, Eliminating Decryption Failures from the Simple Matrix Encryption Scheme, Cryptology ePrint Archive 2016/010.
[POG15] T. Pöppelmann, T. Oder and T. Güneysu, High-Performance Ideal Lattice-Based Cryptography on 8-bit ATxmega Microcontrollers, Cryptology ePrint Archive: Report 2015/382, 2015.
[S97] P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM Journal on Computing, 26(5), pp. 1484-1509, 1997.
[SSH11] K. Sakumoto, T. Shirai and H. Hiwatari, Public-key identification schemes based on multivariate quadratic polynomials, CRYPTO 2011, LNCS 6841, pp. 706 - 723, Springer, 2011.
[SSS10] R. P. Singh, A. Saikia, and B. Sarma, Little Dragon two: an efficient multi-

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

variate public key cryptosystem, International Journal of Network Security & Its Applications, Vol. 2, No. 2, pp. 1-10, 2010.
[SSS11] R. P. Singh, A. Saikia, and B. Sarma, Poly-dragon: an efficient multivariate public key cryptosystem, Journal of Mathematical Cryptology, Vol. 4, pp. 349-364, 2011.
[ST15] W. Shen and S. Tang, RGB, a Mixed Multivariate Signature Scheme, The Computer Journal, Section D: Security in Computer Systems and Networks, 2015.
[TDTD13] C. Tao, A. Diene, S. Tang, and J. Ding, Simple Matrix Scheme for Encryption, PQCrypto 2013, LNCS 7932, pp. 231-242, 2013.
[TXPD15] C. Tao, H. Xiang, A. Petzoldt, and J. Ding, Simple Matrix - A Multivariate Public Key Cryptosystem (MPKC) for Encryption, Finite Fields and Their Applications, Vol. 35, pp. 352-368, 2015.
[TLS16] S. Tang, B. Lv, and W. Shen, Hybrid MQ Signature for Embedded Device, ACISP 2016, LNCS 9722, pp. 281-290, 2016.
[Tho13] E. Thomae, About the Security of Multivariate Quadratic Public Key Schemes, Dissertation Thesis by Dipl. math., RUB, 2013.
[TW12a] E. Thomae, and C. Wolf, Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited, PKC 2012, LNCS 7293, pp. 156-171, 2012.
[TW12b] E. Thomae, and C. Wolf, Cryptanalysis of Enhanced TTS, STS and All Its Variant, or: Why Cross-Terms are Important, Africacrypt 2012, LNCS 7374, pp. 188-202, 2012.
[WUW13] E. Wenger, T. Unterluggauer, and M. Werner, 8/16/32 shades of elliptic curve cryptography on embedded processors, INDOCRYPT 2013, LNCS 8250, pp. 244 - 261, 2013.
[WP04] C. Wolf, and B. Preneel, Equivalent Keys in Multivariate Quadratic Public Key Systems, Cryptology ePrint Archive 2004/360.
[YC05] B.-Y. Yang and J.-M. Chen, Building Secure Tame-like Multivariate Public-Key Crytposystems: The New TTS, ACISP 2005, LNCS 3574, pp. 518-531, Springer, 2005.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

국가연구개발 보고서원문 성과물 전담기관인 한국과학기술정보연구원에서 가공·서비스 하는 연구보고서는 동의 없이 상업적 및 기타 영리목적으로 사용할 수 없습니다.

주 의

1. 이 보고서는 국가수리과학연구소에서 시행한 주요사업의 최종보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 국가수리과학연구소에서 시행한 2016년도 주요연구사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니 됩니다.