AIMer 표준화 문서 작성 방안

2025년 7월

김성광

삼성SDS

AIMer의 구조

AIMer 키 생성

1. pt, iv 생성
2. AIM2$(\mathsf{pt}, \mathsf{iv}) = \mathsf{ct}$ 계산
3. $\mathsf{sk} = (\mathsf{pt}, \mathsf{iv}, \mathsf{ct})$, $\mathsf{pk} = (\mathsf{iv}, \mathsf{ct})$

AIMer 서명 생성의 큰 구조

- 5단계의 Challenge-Response 구조
- Challenge는 Fiat-Shamir transform을 이용하여 해시 함수를 통해 생성
- Phase 1,3,5는 τ 번의 반복이 있으며, 각 반복마다 N 개의 파티에 대한 계산이 필요함
- Phase 2,4에서는 τ 개의 response를 한 번에 해시하여 하나의 해시값을 생성, 이후 필요한만큼의 challenge로 확장함

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
    - GGM 트리를 이용하여 각 파티의 시드 생성
    - 각 파티의 시드를 PRG에 넣고 랜덤 뷰를 생성
    - 랜덤 뷰와 실제 값과 차이를 보정
    - 시드를 커밋 (해시 함수 계산)

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
2. 커밋값과 보정값을 해시 함수에 넣어 첫 번째 챌린지 생성

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
2. 커밋값과 보정값을 해시 함수에 넣어 첫 번째 챌린지 생성
3. 각각의 뷰가 곱셈을 잘 진행하는지 체크 (Multiplication check)
    - SPDZ의 sacrificing technique을 통해 multiplication triple과 뷰를 동시에 체크
    - Multiplication check의 output share를 커밋

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
2. 커밋값과 보정값을 해시 함수에 넣어 첫 번째 챌린지 생성
3. 각각의 뷰가 곱셈을 잘 진행하는지 체크 (Multiplication check)
4. output share 값을 해시하여 두 번째 챌린지 생성

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
2. 커밋값과 보정값을 해시 함수에 넣어 첫 번째 챌린지 생성
3. 각각의 뷰가 곱셈을 잘 진행하는지 체크 (Multiplication check)
4. output share 값을 해시하여 두 번째 챌린지 생성
5. 챌린지가 아닌 나머지 파티의 시드를 전부 공개 (GGM copath)

AIMer 서명 생성 프로세스

0. AIM2 인스턴스를 생성
1. 각 파티의 뷰를 생성하고 커밋
2. 커밋값과 보정값을 해시 함수에 넣어 첫 번째 챌린지 생성
3. 각각의 뷰가 곱셈을 잘 진행하는지 체크 (Multiplication check)
4. output share 값을 해시하여 두 번째 챌린지 생성
5. 챌린지가 아닌 나머지 파티의 시드를 전부 공개 (GGM copath)
6. 서명은 (나머지 시드, 공개 안된 커밋값, 보정값, 챌린지, mult check 관련 마스킹값)

AIMer 서명 검증의 큰 구조

- 서명 생성의 5단계 중 마지막 단계를 제외하고 다시 계산하여 2번째 챌린지가 같은지 확인하는 구조
- 공개 안된 커밋값들은 서명에서 활용하여 계산

AIMer 서명 검증 프로세스

- AIM2 인스턴스를 생성
- Phase 1&2 다시 계산
  - GGM copath를 이용하여 각 반복 별 공개된 시드를 복구
  - 공개된 시드를 통해 공개된 파티의 뷰를 복구 (보정값 포함)
  - 공개된 시드의 커밋을 계산하고, 가려진 시드의 커밋을 받아 첫 번째 챌린지 계산

AIMer 서명 검증 프로세스

- AIM2 인스턴스를 생성
- Phase 1&2 다시 계산
  - GGM copath를 이용하여 각 반복 별 공개된 시드를 복구
  - 공개된 시드를 통해 공개된 파티의 뷰를 복구 (보정값 포함)
  - 공개된 시드의 커밋을 계산하고, 가려진 시드의 커밋을 받아 첫 번째 챌린지 계산
- Phase 3&4 다시 계산
  - 공개된 파티에 대하여 multiplication check 프로토콜을 계산하고 mult check에 대한 커밋값 복구
  - 가려진 커밋값과 마스킹값을 받아 두번째 챌린지 계산

AIMer 서명 검증 프로세스

- AIM2 인스턴스를 생성
- Phase 1&2 다시 계산
  - GGM copath를 이용하여 각 반복 별 공개된 시드를 복구
  - 공개된 시드를 통해 공개된 파티의 뷰를 복구 (보정값 포함)
  - 공개된 시드의 커밋을 계산하고, 가려진 시드의 커밋을 받아 첫 번째 챌린지 계산
- Phase 3&4 다시 계산
  - 공개된 파티에 대하여 multiplication check 프로토콜을 계산하고 mult check에 대한 커밋값 복구
  - 가려진 커밋값과 마스킹값을 받아 두번째 챌린지 계산
- 두번째 챌린지가 같으면 ACCEPT, 다르면 REJECT

기초 함수들

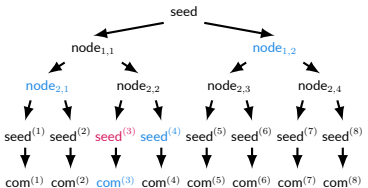
- 유한체 곱연산

기초 함수들

- 유한체 곱연산
- 해시 함수
  - 도메인이 분리된 해시 함수들($H_0, \ldots, H_5$)
  - 챌린지 생성 함수 (ExpandH1, ExpandH2)
  - IV 확장 함수 (ExpandIV)
  - 모두 SHAKE 사용

기초 함수들

- 유한체 곱연산
- 해시 함수
- GGM 트리 연산 함수
  - ExpandTree: 루트 시드로부터 GGM 트리를 생성하는 과정
  - RevealAllBut: 하나의 시드를 제외한 나머지 시드를 공개하는 copath를 계산하는 과정
  - ReconstructSeedTree: 주어진 copath로부터 punctured tree를 복구하는 과정



기초 함수들

- 유한체 곱연산
- 해시 함수
- GGM 트리 연산 함수
- AIM2 함수
  - GenerateLinear: AIM2에 쓰이는 선형층을 IV로부터 생성하는 과정 (ExpandIV 이용)
  - AIM2: AIM2 계산
  - AIM2_MPC: witness share를 가지고 AIM2에 해당하는 곱셈 회로 생성

Oversimplified Pseudocode - Sign

Algorithm 1: AIMer Sign

$\mu \leftarrow H_0(\mathsf{iv}, \mathsf{ct}, m)$
$(\mathsf{Lin}, \mathsf{Wit}) \leftarrow \mathsf{InitAIM2}(sk, \mathsf{iv})$
$(\mathsf{salt}, (\mathsf{seed}_k)_{k\in[\tau]}) \leftarrow_{\$} \{0,1\}^{(\tau+1)\lambda}$
for $k \in [\tau]$ **do**
    $\mathsf{seed}_k^{(1)}, \ldots, \mathsf{seed}_k^{(N)} \leftarrow \mathsf{ExpandTree}(\mathsf{salt}, k, \mathsf{seed}_k)$
    for $i \in [N]$ **do** $(\mathsf{com}_k^{(i)}, \mathsf{view}_k^{(i)}) \leftarrow H(\mathsf{salt}, k-1, i-1, \mathsf{seed}_k^{(i)})$
    Compute corrections corr_k and adjust last shares
end
$\mathsf{chall}_1 \leftarrow H(\mu, \sigma_1, \mathsf{salt}, \mathsf{com}, (\mathsf{corr}_k)_{k\in[\tau]})$
for $k \in [\tau]$ **do** $\mathsf{multCom}_k \leftarrow \mathsf{MultCheck}(\mathsf{view}_k, \mathsf{chall}_1)$
$\mathsf{chall}_2 \leftarrow H(\mathsf{chall}_1, \mathsf{salt}, \mathsf{multCom})$
for $k \in [\tau]$ **do** $\mathsf{copath}_k \leftarrow \mathsf{RevealAllBut}(\mathsf{nodes}_k, \mathsf{chall}_2)$
Return $\sigma \leftarrow \left(\mathsf{salt}, \mathsf{chall}_1, \mathsf{chall}_2, (\mathsf{copath}_k, \mathsf{com}_k^{(\bar{i}_k)}, \mathsf{corr}_k, \mathsf{multCom}_k)_{k\in[\tau]}\right)$

Oversimplified Pseudocode - Verify

Algorithm 2: AIMer Verify

$\mu \leftarrow H_0(\mathsf{iv}, \mathsf{ct}, m)$
$\mathsf{Lin} \leftarrow \mathsf{GenerateLinear}(\mathsf{iv})$
for $k \in [\tau]$ **do**
    $\mathsf{seed}_k^{(1)}, \ldots, \mathsf{seed}_k^{(N)} \leftarrow \mathsf{ReconstructSeedTree}(\mathsf{salt}, \mathsf{copath}_k, k, \bar{i}_k)$
    for $i \in [N] \setminus \{\bar{i}_k\}$ **do** $(\mathsf{com}_k^{(i)}, \mathsf{view}_k^{(i)}) \leftarrow H(\mathsf{salt}, k-1, i-1, \mathsf{seed}_k^{(i)})$
    Adjust last shares using corr_k
end
$\mathsf{chall}_1' \leftarrow H(\mu, \sigma_1, \mathsf{salt}, \mathsf{com}, (\mathsf{corr}_k)_{k \in [\tau]})$
for $k \in [\tau] \setminus \{\bar{i}_k\}$ **do** $\mathsf{multCom}_k \leftarrow \mathsf{MultCheck}(\mathsf{view}_k, \mathsf{chall}_1)$
$\mathsf{chall}_2' \leftarrow H(\mathsf{chall}_1, \mathsf{salt}, \mathsf{multCom})$
Return Accept if $\mathsf{chall}_1 = \mathsf{chall}_1'$ and $\mathsf{chall}_2 = \mathsf{chall}_2'$ or Reject otherwise

표준 문서 작성 방향

개요

1. 적용범위
2. 인용표준
3. 용어와 정의
4. 기호 및 표기
5. 수학적 규정
6. 일반 모형
7. 기초 함수
8. 서명 프로세스

인용표준

- KS X ISO/IEC 10118: 정보기술 - 보안기술 - 해시함수
- KS X ISO/IEC 14888-1: 정보기술 - 보안기술 - 부가형 디지털 서명 - 제1부: 일반
- AES는 사용하지 않아 인용할 필요 없음

용어와 정의

- 데이터 관련 용어 인용: 데이터 요소, 옥텟, 데이터 열
- 기초 암호학 용어 인용: 키 쌍, 메시지, 서명, 서명 키, 서명 과정, 서명된 메시지, 검증 키, 검증 과정, 솔트, 초기값
- 수학적 용어 설명: 유한체(finite field)
- 해시 함수 관련 용어 인용: 해시 함수, 해시 코드, 충돌 저항 (회피) 해시 함수

기호 및 표기, 수학적 규정

- 기호 및 표기
  - 수학적 기호: XOR, summation 등
  - 코딩 관습: big/little-endian
- 수학적 규정
  - 유한체 곱셈 과정 설명
  - 행렬 곱셈 과정 설명

서명 프로세스

- 일반 모형: KS X ISO/IEC 14888에 편입된다면 필요 없음
- 기초 함수
- 서명/검증 프로세스

고려할 점 및 의문점

고려할 점

- 서명의 경우 형상이 서로 달라 다른 문서로 진행하는 것이 좋을 것으로 보임

의문점

- SHAKE or SHA3 표준 부재 (ISO/IEC 표준에는 있는 것으로 보임)
- KS X ISO/IEC 14888 표준의 하위 표준으로 진행해야할 것인가?

Thank you!
Check out our website!