MBC Lab.

KONKUK UNIVERSITY

# CMBC - ARIA

MBC LAB
윤상준, 허주승

# Index

- 검증 목표
- 검증 내용 – **C Course Code**
- 검증 결과 – **ARIA Performance Test Using CBMC**
- 결론

# 검증 목표

- **ARIA**의 단계별 **CBMC**를 통한 검증
  - ▶라운드 키 생성
    - 입력 값에 대한 목표 개수의 라운드 키 생성
  - ▶치환계층
    - 치환 이후 목표한 값이 정확히 나오는지 검증
    - **S-box**의 키교환이 정확히 이루어 지는지 검증
  - ▶확산계층
    - 암호화 알고리즘 각 **bit**별 목표로 하는 **bit** 수 확인

검증 내용

# C SOURCE CODE

# 라운드 키 생성

- **C source code** 입력
  - 각 **bit**별 생성 **Round key** 개수 확인

```
Aria_RotXOR (w0, 0, e      ); Aria_RotXOR (w1,  19, e      );
assert( e != ((void *)0) );
Aria_RotXOR (w1, 0, e +  16); Aria_RotXOR (w2,  19, e +  16);
assert( e != ((void *)0) );
Aria_RotXOR (w2, 0, e +  32); Aria_RotXOR (w3,  19, e +  32);
assert( e != ((void *)0) );
Aria_RotXOR (w3, 0, e +  48); Aria_RotXOR (w0,  19, e +  48);
assert( e != ((void *)0) );
Aria_RotXOR (w0, 0, e +  64); Aria_RotXOR (w1,  31, e +  64);
assert( e != ((void *)0) );
Aria_RotXOR (w1, 0, e +  80); Aria_RotXOR (w2,  31, e +  80);
assert( e != ((void *)0) );
Aria_RotXOR (w2, 0, e +  96); Aria_RotXOR (w3,  31, e +  96);
assert( e != ((void *)0) );
```

```
if(keyBits == 256)
      assert(R == 16);
else if(keyBits == 192)
      assert(R == 14);
else if(KeyBits == 128)
      assert(R == 12);
 else
      assert(0);
```

- **C source code** 입력
  - 예제로 나온 **Aria**의 치환 이후 목표한 값이 정확히 나오는지 검증

```
// Right-rotate 128 bit source string s by n bits and XOR it to target string t
void Aria_RotXOR (const unsigned char *s, unsigned int n, unsigned char *t)
{
   unsigned char i, q;

   q = r/8; n %= 8;
   for (i = 0; i < 16; i++) {
      t[(q+i) % 16] ^= (s[i] >> n);
      if (n != 0) t[(q+i+1) % 16] ^= (s[i] << (8-n));
   }
/* 행이동(오른쪽 시프트연산) 과정중 s 와 t의 값이 정상적인지 비교 */
   assert( s != ((void *)0) );
   assert( t != ((void *)0) );
}
```

- **C source code 입력**
  - **S-box**의 키교환이 정확히 이루어 지는지 검증

```
for (j = 0; j < 16; j++) c[j] = p[j];
for (i = 0; i < R/2; i++)
{
    for (j = 0; j < 16; j++) t[j] = S[      j  % 4][e[j] ^ c[j]];
    Aria_DL(t, c); e += 16;
    for (j = 0; j < 16; j++) t[j] = S[(2 + j) % 4][e[j] ^ c[j]];
    Aria_DL(t, c); e += 16;
}
assert( t != ((void *)0) );
assert( c != ((void *)0) );
```

# 확산 계층

- **C source code** 입력
  - 모든 **ARIA bit**별로 목표 **bit**수가 생기는지 확인
  - **128, 192, 256** 비트 별 목표 라운드 키의 비트 확인 및 개수 확인.

```
if(nKeyBit == 256) {
   for(i = 0; ;i++) {
      if( i <= 16)
         assert( *rk[i] != 0x00 );
      else {
         assert( *rk[i] == 0x00 );
            break;
      }
   }
}
else if(nKeyBit == 192) {
   for(i = 0; ;i++) {
      if( i <= 14)
         assert( *rk[i] != 0x00 );
      else {
         assert( *rk[i] == 0x00 );
         break;
      }
   }
}
```

```
else if(nKeyBit == 128) {
   for(i = 0; ;i++) {
      if( i <= 12)
         assert( *rk[i] != 0x00 );
      else {
         assert( *rk[i] == 0x00 );
         break;
      }
   }
}
```

검증 결과

# ARIA PERFORMANCE TEST USING CBMC

Mobile Broadband Communication Lab., Konkuk University

# 라운드 키 생성

실패시 결과

```
State 2729 file aria.c line 119 function Aria_RotXOR thread 0
------------------------------------------------
  Aria_RotXOR::1::i=16 (00010000)

Violated property:
  file aria.c line 165 function Aria_EncKeySetup
  assertion
  e == NULL

VERIFICATION FAILED
```

**TEST 결과**
**SUCCESSFUL**

```
size of program expression: 5227 assignments
simple slicing removed 3 assignments
Generated 86 VCC(s), 38 remaining after simplification
Passing problem to propositional reduction
Running propositional reduction
Solving with MiniSAT2 without simplifier
34358 variables, 54802 clauses
SAT checker inconsistent: negated claim is UNSATISFIABLE, i.e., holds
Runtime decision procedure: 4.399s
VERIFICATION SUCCESSFUL
```

# 치환 계층

실패시 결과

```
State 7782 file aria.c line 241 function Aria_Crypt thread 0
------------------------------------------------
  Aria_Crypt::1::j=16 (00000000000000000000000000010000)

Violated property:
  file aria.c line 243 function Aria_Crypt
  assertion
  &t[0] == NULL

VERIFICATION FAILED
```

**TEST 결과**
**SUCCESSFUL**

```
size of program expression: 6506 assignments
simple slicing removed 3 assignments
Generated 114 VCC(s), 38 remaining after simplification
Passing problem to propositional reduction
Running propositional reduction
Solving with MiniSAT2 without simplifier
191558 variables, 1095182 clauses
SAT checker inconsistent: negated claim is UNSATISFIABLE, i.e., holds
Runtime decision procedure: 9.68s
VERIFICATION SUCCESSFUL
```

# 확산 계층

실패시 결과

```
State 6320 file aria.c line 241 function Aria_Crypt thread 0
------------------------------------------------
  Aria_Crypt::1::j=16 (00000000000000000000000000010000)

State 6326 file aria.c line 289 function main thread 0
------------------------------------------------
  main::1::i=0 (00000000000000000000000000000000)

Violated property:
  file aria.c line 291 function main
  assertion
  (int)(rk[(int)i][0]) == 0

VERIFICATION FAILED
```

**TEST 결과**
**SUCCESSFUL**

```
size of program expression: 6506 assignments
simple slicing removed 3 assignments
Generated 114 VCC(s), 38 remaining after simplification
Passing problem to propositional reduction
Running propositional reduction
Solving with MiniSAT2 without simplifier
191558 variables, 1095182 clauses
SAT checker inconsistent: negated claim is UNSATISFIABLE, i.e., holds
Runtime decision procedure: 9.68s
VERIFICATION SUCCESSFUL
```

# 수정한 ARIA Code Source에서 TEST시 문제점

- **ARIA TEST**에서 **Pointer Check**시**Aria_Crypt** 함수에서 **FAILED**이 출력되었음.

```
State 8000 file aria.c line 231 function Aria_Crypt thread 0
------------------------------------------------
  Aria_Crypt::1::j=5 (00000000000000000000000000000101)

Violated property:
  file aria.c line 231 function Aria_Crypt
  dereference failure: object upper bound
  !(POINTER_OFFSET(p) + (int)j > 4) || !(SAME-OBJECT(p, &"test"[0]))

VERIFICATION FAILED

F:\tool\cbmc>cbmc aria.c --function main --pointer-check
```

- **Pointer** 수정 후 및 기존의 **ARIA Source Code Test**시 문제가 발생하지 않았음

# ARIA 암호화 알고리즘 CBMC 검증결과

| | **ARIA Source** | 수정한 **ARIA Source** |
|---|---|---|
| Bounds Check | SUCCESSFUL | SUCCESSFUL |
| Pointer Check | SUCCESSFUL | VERIFICATION FAILED<br>→ Code 수정 후 성공 |
| Overflow Check | SUCCESSFUL | SUCCESSFUL |

# 다른 암호화 검증비교 목표

- **ARIA** 이외 **128bit**의 암호화 알고리즘별 검증
- 동일한 평문 키를 암호화 시키는 작동 시간 비교
- 비교 암호화 알고리즘
  - ►**3DES**

# 암호화 알고리즘 TEST

- **ARIA, DES** 결과 비교
  - ▶**ARIA** 검증 결과
    - **Bounds check** 결과 이상 없었으며, 검증처리시간은 약 **30s** 정도 소모되었음

```
Unwinding loop c::main.5 iteration 13 file aria.c line 289 function main
size of program expression: 6986 assignments
simple slicing removed 3 assignments
Generated 1216 VCC(s), 518 remaining after simplification
Passing problem to propositional reduction
Running propositional reduction
Solving with MiniSAT2 without simplifier
207702 variables, 1134091 clauses
SAT checker: negated claim is UNSATISFIABLE, i.e., holds
Runtime decision procedure: 30s
VERIFICATION SUCCESSFUL

F:₩tool₩cbmc>cbmc aria.c --function main --bounds-check
```

- **ARIA, 3DES** 결과 비교
  - ▶**3DES** 검증 결과
    - **Bounds check** 결과 **1~2**시간 이상 되는 시간이 소모되었으며, **FAILED** 출력

```
State 10003 file 3des.c line 620 function Compress_P_Box thread 0
------------------------------------------------
  Compress_P_Box::1::tempbit=0 (00000000000000000000000000000000)

State 10004 file 3des.c line 621 function Compress_P_Box thread 0
------------------------------------------------
  Compress_P_Box::1::mask=2147483648 (10000000000000000000000000000000)

Violated property:
  file 3des.c line 623 function Compress_P_Box
  array `PC2' upper bound
  28 + i < 48

VERIFICATION FAILED

F:₩tool₩cbmc>cbmc 3des.c --function main --bounds-check
```

# 결론

- **ARIA**의 **CBMC Test** 결과
  - **ARIA.C** 에 추가의 **assert** 문을 추가하였음.
  - **Assert** 문이 **Error** 발생시 정상적으로 작동하는지 **Test**하였음.
  - **CBMC**를 이용한 **ARIA.c** 검증결과 모든 함수에 대해서 정상적으로 작동하였음.
  - 라운드 키 생성, 치환계층, 확산계층의 동작에 문제가 없음을 알 수 있음.

- 암호 알고리즘 **3DES**와의 비교 결과
  - **3DES**의 경우 매우 긴 검증시간이 소모되었으며, **Bounds Check**시 **FAILED**이 출력되었음.

# 향후 계획

- **Sensor Network** 및 **Ad-Hoc Network**에서 암호화시 **AIRA** 암호화 사용에 따른 성능 향상 **Test**

- **IPsec Protocol**의 다양한 알고리즘 **Test**
  - **NS-2** 등 네트워크 시뮬레이션에서도 동일한 결과 비교

# 감사합니다.