Réunion flash

Point hebdomadaire

Duzés Florian

Sommaire

01
État des lieux

Point actuel

Fait → **En cours** → **Prévus**
Pour le 15/06

Fait

- Méthodologie x86_64
- Méthodologie ARM
- Schémas de compilation

En cours

- Automatisation des *-test.c*
- Préparation d'une config pour la génération des tests
- Vérification de **p256** en ARM

Prévus

- Automatisation des .ini
- Couvrir les tests
- Couvrir les primitives Hacl*
- RiskV - compil à la main de tests

Réalisation

Fait

- Chaîne de compilation
- Identifications d'erreur de **p256**

En cours

- Automatisation des *-test.c*
- Préparation d'une config pour la génération des tests

Prévus
Pour le 15/06

- Automatisation des .ini
- Couvrir les tests
- Couvrir les primitives Hacl*
- RiskV - compil à la main de tests

02
Couverture de Hacl*

Fabrication des tests

Premiers scripts pondus

La semaine dernière

```
uint32_t
  Hacl_AEAD_Chacha20Poly1305_decrypt
  uint8_t *output,   uint8_t *input,
  uint32_t input_len,   uint8_t *data,
  uint32_t data_len,   uint8_t *key,
  uint8_t *nonce,   uint8_t *tag
```

Code – Hacl_AEAD_Chacha20Poly1305_decrypt

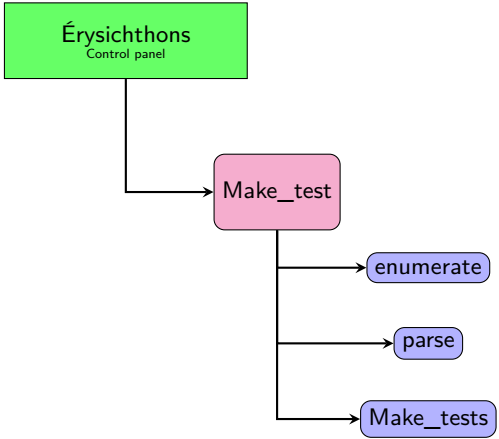
Fabrication des tests

Premiers scripts pondus

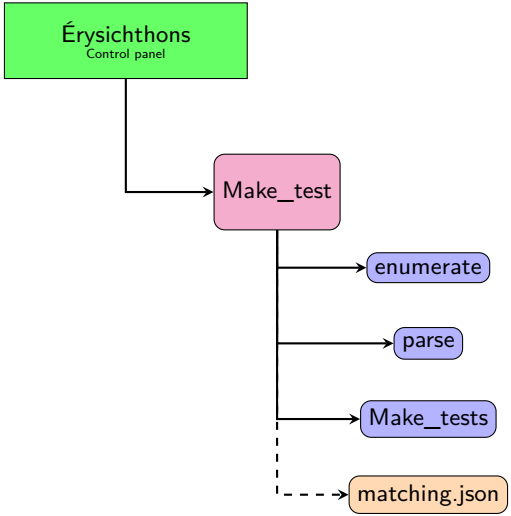
```c
#include <stdlib.h>
#include Hacl_AEAD_Chacha20Poly1305.h
#define BUF_SIZE 16384
#define KEY_SIZE 32
#define TAG_SIZE 16
#define AAD_SIZE 12
#define NONCE_SIZE 12
uint8_t output[BUF_SIZE];
uint8_t input[BUF_SIZE];
uint8_t data[AAD_SIZE];
uint8_t key[KEY_SIZE];
uint8_t nonce[NONCE_SIZE];
uint8_t tag[TAG_SIZE];
int main (int argc, char *argv[]){
uint32_t a =    Hacl_AEAD_Chacha20Poly1305_decrypt
      (output, input, BUF_SIZE, data, AAD_SIZE, key, nonce, tag);
  exit(0);
}
```

Code – Hacl_AEAD_Chacha20Poly1305_decrypt.c

Chaîne de fabrication



Chaîne de fabrication



Phase de transformation

```
uint32_t
  Hacl_AEAD_Chacha20Poly1305_decrypt
  uint8_t *output,   uint8_t *input,
  uint32_t input_len,   uint8_t *data,
  uint32_t data_len,   uint8_t *key,
  uint8_t *nonce,   uint8_t *tag
```

Code – Hacl_AEAD_Chacha20Poly1305_decrypt

Phase de transformation

```
uint32_t
  Hacl_AEAD_Chacha20Poly1305_decrypt
  uint8_t *output,   uint8_t *input,
  uint32_t input_len,   uint8_t *data,
  uint32_t data_len,   uint8_t *key,
  uint8_t *nonce,   uint8_t *tag
```

Code – Hacl_AEAD_Chacha20Poly1305_decrypt

```
{
  "input":"BUF_SIZE",
  "input_len":"BUF_SIZE",
  "output":"BUF_SIZE",
  "output_len":"BUF_SIZE",
  "key":"KEY_SIZE",
  "nonce":"NONCE_SIZE",
  "tag":"TAG_SIZE",
  "data":"AAD_SIZE",
  "data_len":"AAD_SIZE",
  "BUF_SIZE":16384,
  "KEY_SIZE":32,
  "NONCE_SIZE":12,
  "AAD_SIZE":12,
  "TAG_SIZE":16
}
```

Code – matching.json

Phase de transformation

```
uint32_t
  Hacl_AEAD_Chacha20Poly1305_decrypt
  uint8_t *output,   uint8_t *input,
  uint32_t input_len,   uint8_t *data,
  uint32_t data_len,   uint8_t *key,
  uint8_t *nonce,   uint8_t *tag
```

Code – Hacl_AEAD_Chacha20Poly1305_decrypt

```
{
  "input":"BUF_SIZE",
  "input_len":"BUF_SIZE",
  "output":"BUF_SIZE",
  "output_len":"BUF_SIZE",
  "key":"KEY_SIZE",
  "nonce":"NONCE_SIZE",
  "tag":"TAG_SIZE",
  "data":"AAD_SIZE",
  "data_len":"AAD_SIZE",
  "BUF_SIZE":16384,
  "KEY_SIZE":32,
  "NONCE_SIZE":12,
  "AAD_SIZE":12,
  "TAG_SIZE":16
}
```

Code – matching.json

Hacl_AEAD_Chacha20Poly1305_decrypt.c

03
Reproduction de bug

Etude de ARM

État de l'art

- Compilation des tests

Etude de ARM

État de l'art

- Compilation des tests
- Analyse Binsec difficile sur *p256-test.c*

Etude de ARM

État de l'art

- Compilation des tests
- Analyse Binsec difficile sur *p256-test.c*
  - Simplification des tests
  - Remonté d'une erreur au sein même de la fonction

À la poursuite de l'erreur

Cible :

```
static void cmovznz4(uint64_t cin,
    uint64_t *x, uint64_t *y, uint64_t
    *r)
{
  uint64_t mask = ~FStar_UInt64_eq_mask
      (cin, (uint64_t)0U);
  uint64_t r0 = (y[0U] & mask) | (x[0U]
      & ~mask);
  uint64_t r1 = (y[1U] & mask) | (x[1U]
      & ~mask);
  uint64_t r2 = (y[2U] & mask) | (x[2U]
      & ~mask);
  uint64_t r3 = (y[3U] & mask) | (x[3U]
      & ~mask);
  r=[r0,r1,r2,r3];
}
```

Code – Hacl_P256.h/cmovznz4

```
static inline void bn_cmovznz4(uint64_t
     *res, uint64_t cin, uint64_t *x,
    uint64_t *y)
{
  uint64_t mask = ~FStar_UInt64_eq_mask
      (cin, 0ULL);
  KRML_MAYBE_FOR4(i,
    0U,
    4U,
    1U,
    uint64_t *os = res;
    uint64_t uu____0 = x[i];
    uint64_t x1 = uu____0 ^ (mask & (y[
        i] ^ uu____0));
    os[i] = x1;);
}
```

Code – Hacl_P256.h/bn_cmovznz4

Prise en main de Clang+LLVM

```
Documents
└── cross_compilation
    ├── arm-gnu-toolchain-12.2.rel1-x86_64-arm-
    │   none-linux-gnueabihf
    ├── clang+llvm-14.0.6-x86_64-linux-gnu-rhel-
    │   8.4
    ├── clang+llvm-15.0.6-x86_64-linux-gnu-ubuntu-
    │   18.04
    ├── clang+llvm-16.0.4-x86_64-linux-gnu-ubuntu-
    │   22.04
    ├── clang+llvm-17.0.6-x86_64-linux-gnu-ubuntu-
    │   22.04
    └── clang+llvm-18.1.8-x86_64-linux-gnu-ubuntu-
        18.04
```

```
define compile
$(BUILD_DIR)/$(version)/%.exe: $(SRC_DIR)/%.c
  @mkdir -p $$(dir $$@)
  $(COMPIL)/$(version)/bin/clang $(ARCHI) $(
      CFLAGS) $(FORCE) -c $$< -o $$(patsubst %.
      exe,%.o,$$@)
  $(COMPIL)/$(version)/bin/clang $(ARCHI) $(
      CFLAGS) $(LDFLAGS) $(FORCE) $$(patsubst
      %.exe,%.o,$$@) -static -o $$@
  @rm $(BUILD_DIR)/$(version)/*.o
  @cp binsec_script/$(INI)/* $(BUILD_DIR)/$(
      version)/
endef
```

Code – Makefile

Résultats

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-O3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Os	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Oz	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Résultats

–target=aarch64-none-linux-gnu

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-O3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Os	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Oz	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Résultats

–target=aarch64-none-linux-gnu

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-O3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Os	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Oz	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Armv7

–target=gcc-arm-none-eabi -mcpu=cortex-a9 -marm

Résultats Armv7

–target=armv7-none-linux-gnueabihf

Résultats Armv7

–target=armv7-none-linux-gnueabihf

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	~	~	~	~	~	~	~	~
-O3	✓	✓	~	~	~	~	~	~	~	~
-Os	✓	✓	~	~	~	~	~	~	~	~
-Oz	~	~	~	~	~	~	~	~	~	~

Focus on 15.0.6

Fast

```c
static uint64_t FStar_UInt64_eq_mask(
    uint64_t a, uint64_t b)
{
  uint64_t x = a ^ b;
  uint64_t minus_x = ~x + 1ULL;
  uint64_t x_or_minus_x = x | minus_x;
  uint64_t xnx = x_or_minus_x >> 63U;
  return xnx - 1ULL;
}
```

Code – FStar_UInt64_eq_mask

```c
#include "Hacl_P256.h"
```

Code – FStar_UInt64_eq_mask

Focus on 15.0.6

Fast

```
load sections .text, .rodata, .data, .got, .bss from file

secret global r, cin, y, x


@[sp, 4] := 0x0000000000075570 as fin
starting from <main>
with concrete stack pointer
halt at fin
explore all
```

Code – study.ini

Binsec

```
[sse:info] SMT queries
             Preprocessing simplifications
               total          1
               true           0
               false          0
               constant enum  1

             Satisfiability queries
               total          28
               sat            2
               unsat          26
               unknown        0
               time           0.57
               average        0.02

           Exploration
             total paths                      4
             completed/cut paths              0
             pending paths                    0
             stale paths                      4
             failed assertions                0
             branching points                 1
             max path depth                   24
             visited instructions (unrolled)  24
             visited instructions (static)    24

[checkct:result] Program status is : unknown (0.678)
[checkct:info] 0 visited path covering 24 instructions
[checkct:info] 1 / 1 control flow checks pass
[checkct:info] 29 / 29 memory access checks pass
```

Code – make binsec

Debug

```
[sse:debug] 0x00075560 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x70
[sse:debug] 0x00075564 mov  r0, #0                    # <main> + 0x74
[sse:debug] 0x00075568 ldmia  sp!, {r4, pc}           # <main> + 0x78
[sse:info] Empty path worklist: halting ...
```

Code – FStar_UInt64_eq_mask

Debug

```
[sse:debug] 0x00075560 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x70
[sse:debug] 0x00075564 mov  r0, #0                    # <main> + 0x74
[sse:debug] 0x00075568 ldmia  sp!, {r4, pc}           # <main> + 0x78
[sse:info] Empty path worklist: halting ...
```

Code – FStar_UInt64_eq_mask

```
564:  e3a00000  mov r0, #0
568:  e8bd8010  pop {r4, pc}
56c:  000266c0  andeq r6, r2, r0, asr #13
570:  000266b4      @ <UNDEFINED> instruction: 0x000266b4
```

Code – disas

Debug

```
[sse:debug] 0x00075560 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x70
[sse:debug] 0x00075564 mov  r0, #0                    # <main> + 0x74
[sse:debug] 0x00075568 ldmia  sp!, {r4, pc}           # <main> + 0x78
[sse:info] Empty path worklist: halting ...
```

Code – FStar_UInt64_eq_mask

```
564:  e3a00000  mov r0, #0
568:  e8bd8010  pop {r4, pc}
56c:  000266c0  andeq r6, r2, r0, asr #13
570:  000266b4      @ <UNDEFINED> instruction: 0x000266b4
```

Code – disas

```
<32> := 0x7556c
```

Code – study.ini

Solution finale

```
[checkct:result] Program status is : secure (4.969)
[checkct:info] 1 visited path covering 37 instructions
[checkct:info] 2 / 2 control flow checks pass
[checkct:info] 33 / 33 memory access checks pass
```

Code – study.ini

Binsec - version include

```
Preprocessing simplifications
    total          2
    true           0
    false          0
    constant enum  2

  Satisfiability queries
    total          31
    sat            4
    unsat          27
    unknown        0
    time           4.64
    average        0.15

Exploration
  total paths                       4
  completed/cut paths               0
  pending paths                     0
  stale paths                       4
  failed assertions                 0
  branching points                  2
  max path depth                    36
  visited instructions (unrolled)   36
  visited instructions (static)     36


[checkct:result] Program status is : unknown (4.707)
[checkct:info] 0 visited path covering 36 instructions
[checkct:info] 2 / 2 control flow checks pass
[checkct:info] 33 / 33 memory access checks pass
```

Code – study.ini

Debug - version include

```
[sse:debug] 0x00075544 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x54
[sse:debug] 0x00075548 mov  r0, #0                    # <main> + 0x58
[sse:debug] 0x0007554c bx lr                          # <main> + 0x5c
[sse:info] Empty path worklist: halting ...
```

Code – debug.trace

Debug - version include

```
[sse:debug] 0x00075544 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x54
[sse:debug] 0x00075548 mov  r0, #0                    # <main> + 0x58
[sse:debug] 0x0007554c bx lr                          # <main> + 0x5c
[sse:info] Empty path worklist: halting ...
```

Code – debug.trace

```
75548:  e3a00000  mov r0, #0
7554c:  e12fff1e  bx  lr
75550:  00026694  muleq r2, r4, r6
75554:  00026688  andeq r6, r2, r8, lsl #13
```

Code – FStar_UInt64_eq_mask

Debug - version include

```
[sse:debug] 0x00075544 vst1.64  {d16, d17}, [r0 :128] # <main> + 0x54
[sse:debug] 0x00075548 mov  r0, #0                    # <main> + 0x58
[sse:debug] 0x0007554c bx lr                          # <main> + 0x5c
[sse:info] Empty path worklist: halting ...
```

Code – debug.trace

```
75548:  e3a00000  mov r0, #0
7554c:  e12fff1e  bx  lr
75550:  00026694  muleq r2, r4, r6
75554:  00026688  andeq r6, r2, r8, lsl #13
```

Code – FStar_UInt64_eq_mask

```
lr<32> := 0x75550
```

Code – FStar_UInt64_eq_mask

Solution finale - version include

```
[arm:error] Probable parse error at line 5, column 1
            Lexeme was: undefined
            Entry was: (address . 0x00075550)
(opcode . 0x00026694)
(size . 4)
(mnemonic . "; Unknown ARM instruction")
(undefined)

            Getting basic infos only ...
[sse:error] Cut path 1 (uninterpreted "; Unknown ARM instruction") @ 0x00075550
```

Code – study.ini

Résultats Armv7 - hypothétique

–target=armv7-none-linux-gnueabihf

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-O3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Os	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-Oz	~	~	~	~	~	~	~	~	~	~

Résultats Armv7 - Frais du 22/05

–target=armv7-none-linux-gnueabihf

Clang+LLVM	14.0.6		15.0.6		16.0.4		17.0.6		18.1.8	
opt\src	cmovznz4	bn_cmovznz4	cmov	bn	cmov	bn	cmov	bn	cmov	bn
-O2	✓	✓	✓	~	✓	~	✓	~	✓	~
-O3	✓	✓	✓	~	✓	~	✓	~	✓	~
-Os	✓	✓	✓	~	✓	~	✓	~	✓	~
-Oz	✓	~	✓	~	✓	~	✓	~	✓	~

Sous condition spéciale, voir démo.

04
Conclusion

Conclusion

Automatisation

- Continuer la génération des fichiers *-test.c*
- Activer la chaîne de compilation

Conclusion

Automatisation

- Continuer la génération des fichiers *-test.c*
- Activer la chaîne de compilation

Reproduction de bug

- Observation de nouveaux opcodes
- Sûrement *insecure*

Merci.