Réunion flash

Point hebdomadaire

Duzés Florian

Sommaire

01
État des lieux

Général information

Information exterieur

- Entretien avec Maria Mishtaq
- Réunion Mercredi 2 juillet - 15h30

Point actuel

Fait → **En cours** → **Prévus**

Fait

- Toolchains RiscV
- Tests p256 avec gcc RiscV

En cours

- test p256 avec d'autres compilateurs
- Compiler Hacl* vers RiscV
- Test Binsec sur RiscV
- -test.json

Prévus

- Chaîne de bout en bout
- Couverture des architectures différentes
- Couverture des compilateurs

Réalisation

Fait → **En cours** → **Prévus**

Fait

- Étendu test de p256 avec LLVM / 32 / 64
- -test.json / generic json
- Hacl* -> RiscV

En cours

- Compiler Hacl* vers RiscV
- Test Binsec sur RiscV
- -test.json

Prévus

- Chaîne de bout en bout
- Couverture des architectures différentes
- Couverture des compilateurs

02
Compilation de RiscV

Préambule

Informations générales

- Construction de la toolchain LLVM
  - *GCC*
  - *LLVM*
  - Architecure 64bits : **–with-arch=rv64gc –with-abi=lp64d**
  - - 32bits : **–with-arch=rv32gc –with-abi=lp32d**
- Niveau d'optimisation testé
  - *-O0*, *-O1*, *-O2*, *-O3*, *-Oz*, *-Os*
- Code analysé de *cmovznz4*

Compilation vers riscV-64

	GCC	CLANG+LLVM
-O0	~	~
-O1	✓	X
-O2	✓	X
-O3	✓	X
-Os	✓	X
-Oz	✓	X

Compilation vers riscV-64

	GCC	CLANG+LLVM
-O0	~	~
-O1	✓	X
-O2	✓	X
-O3	✓	X
-Os	✓	X
-Oz	✓	X

- **-O0** error - Binsec ISA definition
- Clang error - **beqz**

Compilation vers riscV-64

	GCC	CLANG+LLVM
-O0	~	~
-O1	✓	X
-O2	✓	X
-O3	✓	X
-Os	✓	X
-Oz	✓	X

- **-O0** error - Binsec ISA definition
- Clang error - **beqz**
- Passage *InstCombinePass*
- patch : # *pragma clang optimise <off/on>*

Compilation vers riscV-64

	GCC	CLANG+LLVM
-O0	~	~
-O1	✓	✓
-O2	✓	✓
-O3	✓	✓
-Os	✓	✓
-Oz	✓	✓

- **-O0** error - Binsec ISA definition
- Clang error - **beqz**
- Passage *InstCombinePass*
- patch : # *pragma clang optimise <off/on>*

Compilation vers riscV-32

	GCC	CLANG+LLVM
-O0	~	✓
-O1	✓	X
-O2	✓	X
-O3	✓	X
-Os	✓	X
-Oz	✓	X

- **Gcc** error - Binsec ISA definition
- Clang error - **beqz**

Compilation vers riscV-32

	GCC	CLANG+LLVM
-O0	~	✓
-O1	✓	✓
-O2	✓	✓
-O3	✓	✓
-Os	✓	✓
-Oz	✓	✓

- **Gcc** error - Binsec ISA definition
- Clang error - **beqz**

03
Compilation croisé HACL*

HACL* compilation

Cible

- aarch64-none-linux-android
- aarch64-none-linux-gnu
- aarch64-apple-darwin
- aarch64-apple-ios
- x86_64-apple-ios-simulator
- *riscv64-unknown-linux-gnu*

04
Érisychton v2

Architecture reconstruite

Amicalement débogable

```
Érysichton
└── make_test
    ├── source
    │   ├── test1.c
    │   ├── test2.c
    │   └── …
    ├── make_test.py
    ├── utils.py
    ├── config.json
    └── matching.json
```

```
Érysichton
└── andhrimnir (make_test)
    ├── source
    │   ├── test1.c
    │   ├── test1.json
    │   └── …
    ├── config
    │   └── (header).json
    ├── make_test.py
    └── lib
        ├── lexer.py
        ├── backend.py
        ├── utils.py
        └── …
```

Fabrication des json

```
{
"Meta_data":{
    "build" : "17-06-2025",
    "version" : "0.2.0"
}

,"Hacl_Curve25519_64_scalarmult": {
    "*out":""
  ,"*priv":""
  ,"*pub":""
}

,"Hacl_Curve25519_64_secret_to_public": {
    "*pub":""
  ,"*priv":""
}

,"Hacl_Curve25519_64_ecdh": {
    "*out":""
  ,"*priv":""
  ,"*pub":""
}
}
```

Code – Hacl_Curve25519_64.json

Remplissage des json

```
{
"Meta_data":{
    "build" : "13-06-2025",
    "version" : "0.2.0"
}

,"Hacl_AEAD_Chacha20Poly1305_Simd128_encrypt": {
    "*output":"BUF_SIZE"
  ,"*input":"BUF_SIZE"
  ,"input_len":"BUF_SIZE"
  ,"*data":"AAD_SIZE"
  ,"data_len":"AAD_SIZE"
  ,"*key":"KEY_SIZE"
  ,"*nonce":"NONCE_SIZE"
  ,"*tag":"TAG_SIZE"
  ,"BUF_SIZE":16384
  ,"TAG_SIZE":16
  ,"AAD_SIZE":12
  ,"KEY_SIZE":32
  ,"NONCE_SIZE":12
}
```

Code – Hacl_AEAD_Chacha20Poly1305_Simd128.json (1)

```
,"Hacl_AEAD_Chacha20Poly1305_Simd128_decrypt": {
    "*output":"BUF_SIZE"
  ,"*input":"BUF_SIZE"
  ,"input_len":"BUF_SIZE"
  ,"*data":"AAD_SIZE"
  ,"data_len":"AAD_SIZE"
  ,"*key":"KEY_SIZE"
  ,"*nonce":"NONCE_SIZE"
  ,"*tag":"TAG_SIZE"
  ,"BUF_SIZE":16384
  ,"TAG_SIZE":16
  ,"AAD_SIZE":12
  ,"KEY_SIZE":32
  ,"NONCE_SIZE":12
  }
}
```

Code – Hacl_AEAD_Chacha20Poly1305_Simd128.json (2)

Construction des tests

```c
//
// Made by
// ANDHRÍMNIR - 0.2.2
// 17-06-2025
//

#include <stdlib.h>
#include "Hacl_AEAD_Chacha20Poly1305.h"

#define tag TAG_SIZE
#define output BUF_SIZE
#define data AAD_SIZE
#define nonce NONCE_SIZE
#define key KEY_SIZE
#define input BUF_SIZE

#define BUF_SIZE 16384
#define AAD_SIZE 12
#define TAG_SIZE 16
#define NONCE_SIZE 12
#define KEY_SIZE 32

uint8_t output[BUF_SIZE];    uint8_t tag[TAG_SIZE];
uint8_t input[BUF_SIZE];     uint8_t data[AAD_SIZE];
uint8_t key[KEY_SIZE];       uint8_t nonce[NONCE_SIZE];

int main (int argc, char *argv[]){
Hacl_AEAD_Chacha20Poly1305_encrypt(output, tag, input, BUF_SIZE, data, AAD_SIZE, key, nonce);
  exit(0);
}
```

Code – Hacl_AEAD_Chacha20Poly1305_Simd128.json

05
Conclusion

Conclusion

Objectif

Finir le module x86_64.

Conclusion

Objectif

Finir le module x86_64.

- ☐ Remplir les configurations
- ☐ Générer les tests
- ☐ Compiler les tests
- ☐ Analyser les tests

Merci.