

Réunion flash

Point hebdomadaire

Duzés Florian




Sommaire

1. État des lieux
2. Compilation de RiscV
3. Compilation croisé HAACL*
4. Érisychton v2
5. Conclusion

01

État des lieux



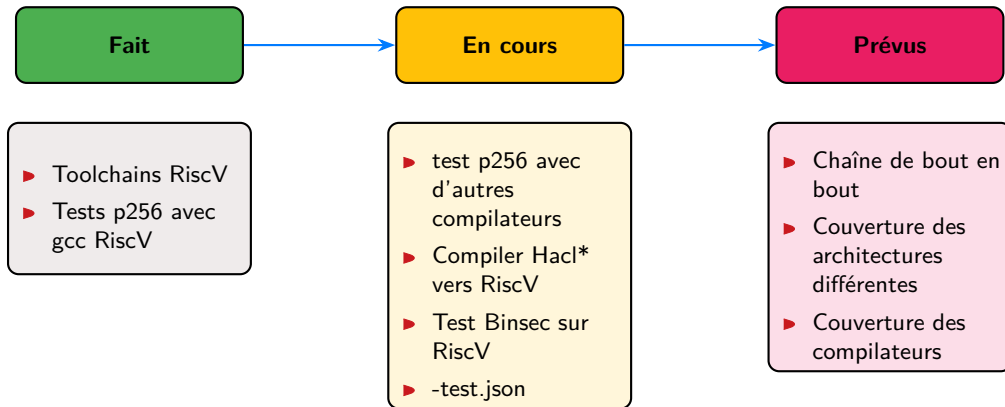


Général information

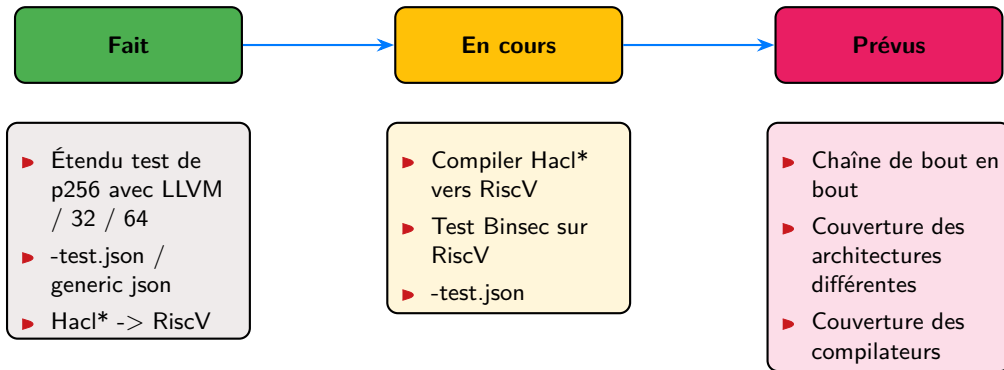
Information extérieur

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

Point actuel



Réalisation



02

Compilation de RiscV



Informations générales

- ▶ Construction de la toolchain LLVM
 - *GCC*
 - *LLVM*
 - Architecture 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

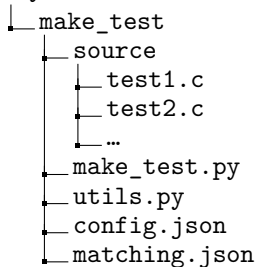




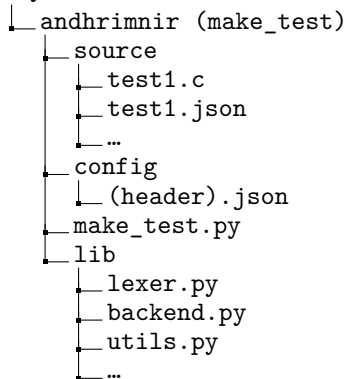
Architecture reconstruite

Amicalement débogable

Érysichton



Érysichton





Fabrication des json

```
1 {
2   "Meta_data":{
3     "build" : "17-06-2025",
4     "version" : "0.2.0"
5   }
6
7   , "Hac1_Curve25519_64_scalarmult": {
8     "*out": ""
9     , "*priv": ""
10    , "*pub": ""
11  }
12
13  , "Hac1_Curve25519_64_secret_to_public": {
14    "*pub": ""
15    , "*priv": ""
16  }
17
18  , "Hac1_Curve25519_64_ecdh": {
19    "*out": ""
20    , "*priv": ""
21    , "*pub": ""
22  }
23 }
```

Code – Hac1_Curve25519_64.json



Remplissage des json

```
1 {
2   "Meta_data":{
3     "build" : "13-06-2025",
4     "version" : "0.2.0"
5   }
6
7   , "Hac1_AEAD_Chacha20Poly1305_Simd128_encrypt": {
8     "*output": "BUF_SIZE"
9     , "*input": "BUF_SIZE"
10    , "input_len": "BUF_SIZE"
11    , "*data": "AAD_SIZE"
12    , "data_len": "AAD_SIZE"
13    , "*key": "KEY_SIZE"
14    , "*nonce": "NONCE_SIZE"
15    , "*tag": "TAG_SIZE"
16    , "BUF_SIZE": 16384
17    , "TAG_SIZE": 16
18    , "AAD_SIZE": 12
19    , "KEY_SIZE": 32
20    , "NONCE_SIZE": 12
21  }
```

Code –

Hac1_AEAD_Chacha20Poly1305_Simd128.json (1)

```
1 , "Hac1_AEAD_Chacha20Poly1305_Simd128_decrypt": {
2   "*output": "BUF_SIZE"
3   , "*input": "BUF_SIZE"
4   , "input_len": "BUF_SIZE"
5   , "*data": "AAD_SIZE"
6   , "data_len": "AAD_SIZE"
7   , "*key": "KEY_SIZE"
8   , "*nonce": "NONCE_SIZE"
9   , "*tag": "TAG_SIZE"
10  , "BUF_SIZE": 16384
11  , "TAG_SIZE": 16
12  , "AAD_SIZE": 12
13  , "KEY_SIZE": 32
14  , "NONCE_SIZE": 12
15  }
16 }
```

Code –

Hac1_AEAD_Chacha20Poly1305_Simd128.json (2)




Construction des tests

```
1  //
2  // Made by
3  // ANDHRÍMNIR - 0.2.2
4  // 17-06-2025
5  //
6
7  #include <stdlib.h>
8  #include "Hac1_AEAD_Chacha20Poly1305.h"
9
10 #define tag TAG_SIZE
11 #define output BUF_SIZE
12 #define data AAD_SIZE
13 #define nonce NONCE_SIZE
14 #define key KEY_SIZE
15 #define input BUF_SIZE
16
17 #define BUF_SIZE 16384
18 #define AAD_SIZE 12
19 #define TAG_SIZE 16
20 #define NONCE_SIZE 12
21 #define KEY_SIZE 32
22
23 uint8_t output[BUF_SIZE];    uint8_t tag[TAG_SIZE];
24 uint8_t input[BUF_SIZE];    uint8_t data[AAD_SIZE];
25 uint8_t key[KEY_SIZE];      uint8_t nonce[NONCE_SIZE];
26
27 int main (int argc, char *argv[]){
28     Hac1_AEAD_Chacha20Poly1305_encrypt(output, tag, input, BUF_SIZE, data, AAD_SIZE, key, nonce);
29     exit(0);
30 }
```

Code – Hac1_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.

