Réunion flash

Point hebdomadaire

Duzes Florian

Sommaire

01
État des lieux

Point actuel

Fait

- Méthodologie x86_64
- Schémas de compilation

En cours

- Méthodologie ARM
- Étendre au dossier */tests*
- Automatisation de x86_64

Prévus
Pour le 15/06

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

Réalisation

Fait

- Méthodologie x86_64
- Méthodologie ARM
- Schémas de compilation

En cours

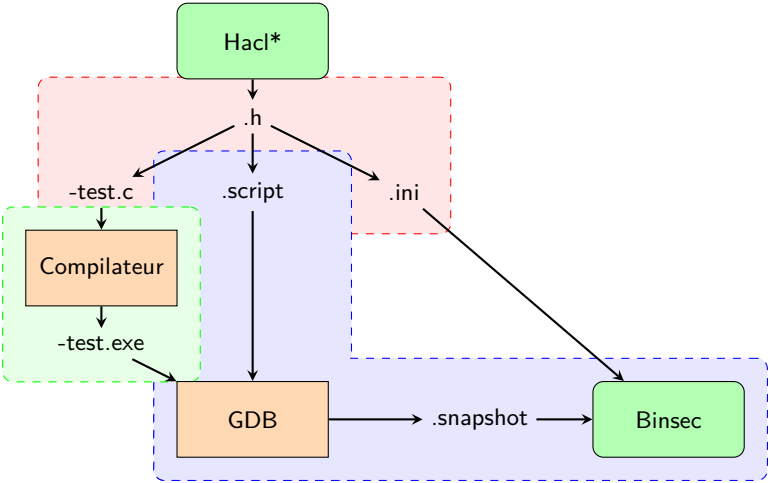
- Automatisation de x86_64
- Conception des fichiers tests

Prévus
Pour le 15/06

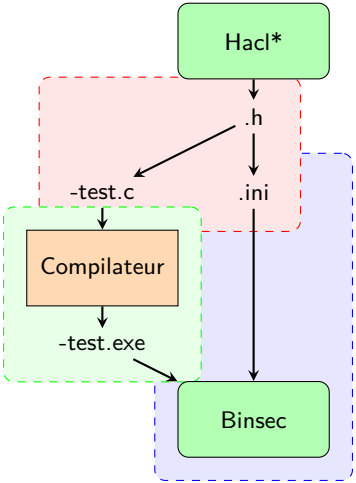
- Automatisation des .ini
- Couvrir les tests
- Couvrir les primitives Hacl*
- RiskV - compil à la main de tests

02
Rappel des arbres

Compilation x86_64

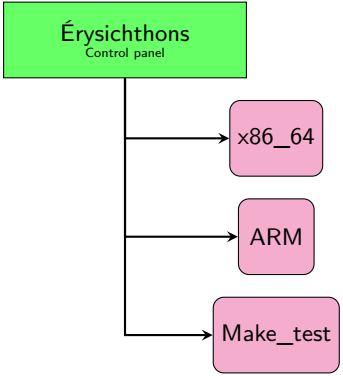


Compilation ARM

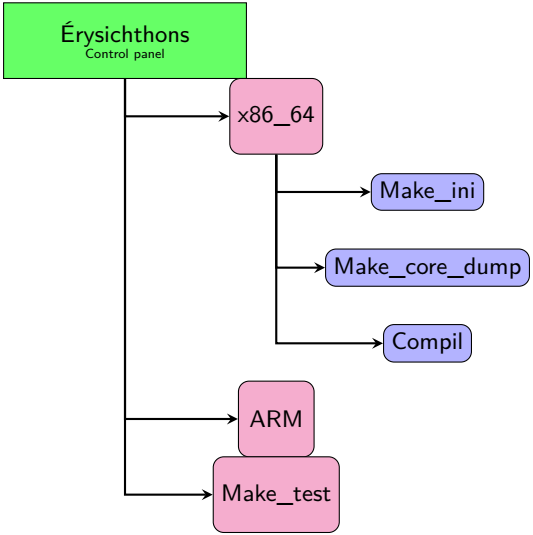


03
Construction de Érysichthon

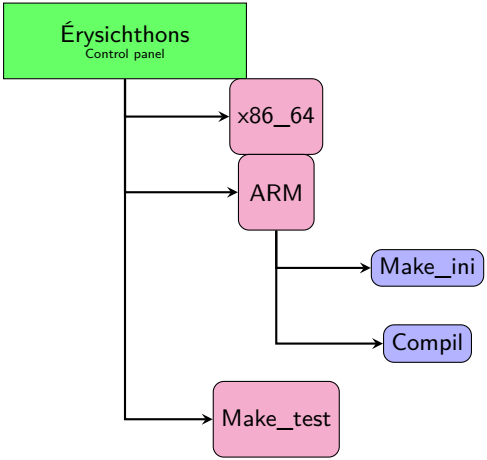
Structure générale



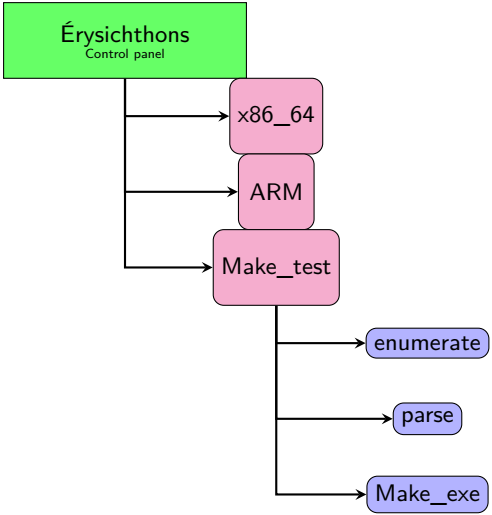
Module x86_64



Module ARM


Érysichthons
Control panel
x86_64
ARM
Make_ini
Compil
Make_test

Module Tests



04
Focus sur le Module Tests

Makefile

```
(DIRECTORY=$(HOME)/Documents/recoules-hacl-star/hacl-star/dist/gcc-compatible
ENUM_FILE=liste
GREP=Hacl

all: enumerate target

enumerate:
  @echo "Enumerate : $(DIRECTORY) - for library $(GREP)"
  @ls $(DIRECTORY)/*.h | grep $(GREP)| xargs -n 1 basename > $(ENUM_FILE)

target:
  @echo "Targeted functions enumerated in : $(ENUM_FILE)"
  @rm -f tests/*
  @echo "Lancement du script\n----" && python3 enum_test.py $(DIRECTORY) $(
      ENUM_FILE)

clean:
  @echo "Nettoyage des fichiers générés..."
  @rm -f $(ENUM_FILE) tests/*
```

Code 1 – Makefile

Résultat courant

```
Hacl_AEAD_Chacha20Poly1305.h
Hacl_AEAD_Chacha20Poly1305_Simd128.h
Hacl_AEAD_Chacha20Poly1305_Simd256.h
Hacl_AES128.h
Hacl_Bignum256_32.h
...
```

Code 2 – liste

```
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 3 – Hacl_AEAD_Chacha20Poly1305-Hacl_AEAD_Chacha20Poly1305_decrypt

```
void
Hacl_AEAD_Chacha20Poly1305_encrypt
  uint8_t *output,   uint8_t *tag,
  uint8_t *input,   uint32_t input_len,
  uint8_t *data,   uint32_t data_len,
  uint8_t *key,   uint8_t *nonce
```

Code 4 – Hacl_AEAD_Chacha20Poly1305-Hacl_AEAD_Chacha20Poly1305_encrypt

Objectif

```
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 5 – Hacl_AEAD_Chacha20Poly1305_decrypt

Code 6 – Hacl_AEAD_Chacha20Poly1305_Simd128_encrypt-test.c

```c
#include <stdlib.h>
#include "Hacl_AEAD_Chacha20Poly1305_Simd128.h"
#define BUF_SIZE 16384
#define KEY_SIZE 32
#define NONCE_SIZE 12
#define AAD_SIZE 12
#define TAG_SIZE 16
uint8_t plain[BUF_SIZE];uint8_t cipher[BUF_SIZE];
uint8_t aead_key[KEY_SIZE];uint8_t aead_nonce[
    NONCE_SIZE];
uint8_t aead_aad[AAD_SIZE];uint8_t tag[16];
int main (int argc, char *argv[]){
  Hacl_AEAD_Chacha20Poly1305_Simd128_encrypt
    (cipher, tag, plain, BUF_SIZE, aead_aad, AAD_SIZE,
        aead_key, aead_nonce);
  exit(0);}
```

05
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
- *Une interface graphique ?*

Merci.