KU LEUVEN

Help choosing the next crypto-standard

ES&S MOL: ay 2022-2023

- ES&S
- Introduction
- Library exercise
- Experiment
- Your turn !

Emerging technologies, Systems & Security

- Research group connected to Electronics/ICT, MNS & COSIC
- Headed by prof. Nele Mentens & prof. Kris Myny
- Current research team:
  - 5 PhD students
  - 1 post-doc
  - 1 research expert

- ES&S
- Introduction
- Library exercise
- Experiment
- Your turn !

Introduction Q1

What **did** you find out about this "formula" ?

cryptology = cryptography + cryptanalysis

Cryptography is used to hide information using ciphers that act as keys to keep the information safe from users who the information is not intended for.[2]

Cryptanalysis is used to reveal the information that was once encrypted. In this process the weaknesses of these cryptographic systems are explored in order to gain access to the encrypted information.[3]

In cryptology both cryptography and cryptanalysis are studied. [1]

Introduction

Q2

What is the difference between Symmetric key and Public key cryptography ?

When the sender and receiver share the same key to scramble and unscramble a message it's called Symmetric Encryption. With Symmetric Encryption, like Caesar Cipher, the secret key has to be agreed on ahead of time by two people in private. So that's great for people, but the internet is open and public so it's impossible for two computers to "meet" in private to agree on a secret key. Instead computers use Asymmetric Encryption keys, a public key that can be exchanged with anybody and a private key that is not shared. The Public Key is used to encrypt data and anybody can use it to create a secret message, but the secret can only be decrypted by a computer with access to the private key.

Public-key cryptography: involves a pair of keys known as public key and a private key. These key pairs are generated with cryptographic algorithms based on mathematical problems.

Public key is available to everyone and it is used to encrypt data. The private key is only known to its owner and it's used to decrypt messages. Public Key cryptography is the foundation of all secure messaging on the open internet.

Introduction Q3

What is a cryptographic algorithm ?

Cryptographic algorithms are sequences of processes used to encrypt and decrypt messages.

Introduction Q3

What is a cryptographic algorithm ?

Symmetric-key cryptography

- DES (Data Encryption Standard)
- 3DES (Triple DES)
- AES (Advanced Encryption Standard)
- Blowfish

Public-key cryptography

- Google: little lock and https (SSL)
- Bitcoin
- SSH
- to send emails (PGP or GPG)

Introduction Q4

Why would you optimise some code or a design towards binary size, **anno 2021** ?

Computers are getting better and better and now the encryption is still safe but in a few years the computers will be able to decrypt it easily.

For example: if you use 10 bit encryption there could be 10 billion possible key solutions. Guessing that key is not difficult for an average computer. It will just take a few seconds to try all 10 billions possibilities.
Today's secure communications are encrypted using 256 bits key to make it "harder" to compute that many possibilities in a reasonable time. The computer chips are getting twice as fast every year so in the near future 256 bit key will not be enough. So you can increase the length of the bits to exponentially increase the number of guesses.

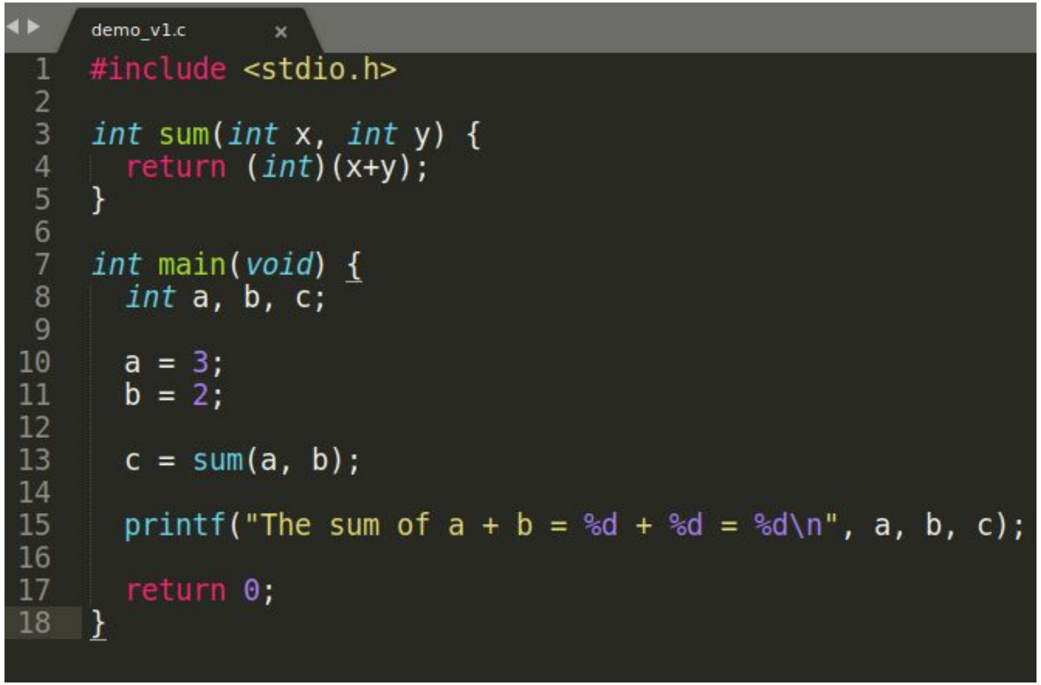
Introduction Q5

How do you compile a static library in C, and how do you link with it ?

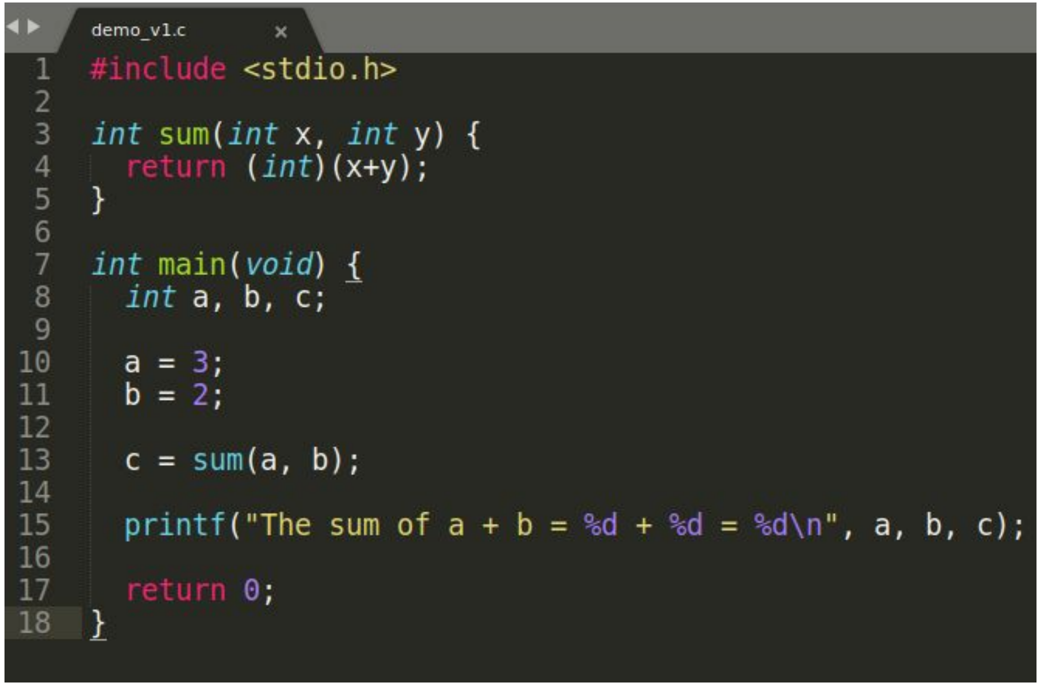
- what is the difference between a static and a dynamic library ?

To compile a static library you need to :
use the -c command so the output is object files that are output by the compiler. Use the ar command to add the generated object files to a static library. Now by adding the libraries name to the object files given to the linker, the static libraries can be used in a program.[4] This can be done using the -L and-l command.

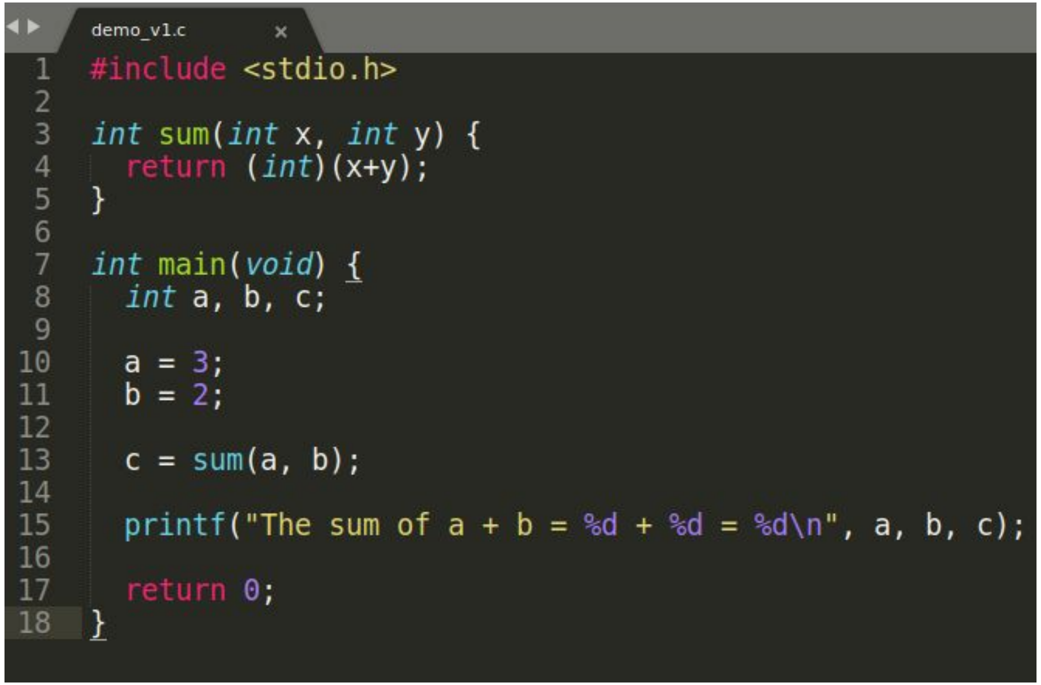
- ES&S
- Introduction
- Library exercise
- Experiment
- Your turn !



```c
#include <stdio.h>

int sum(int x, int y) {
  return (int)(x+y);
}

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```


demo_v1.c

```c
#include <stdio.h>

int sum(int x, int y) {
  return (int)(x+y);
}

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```



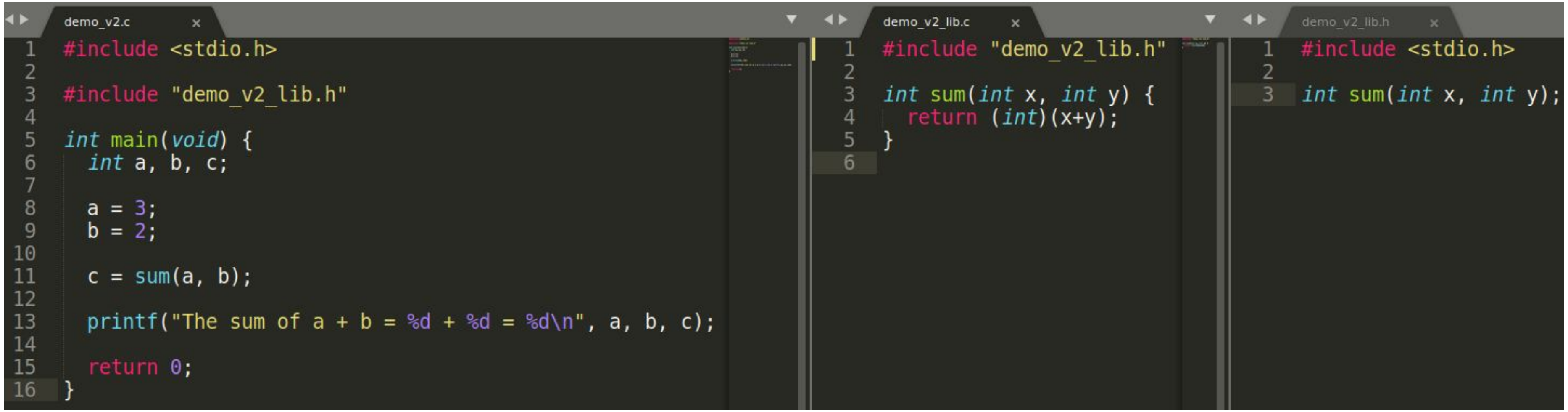
```c
#include <stdio.h>

int sum(int x, int y) {
	return (int)(x+y);
}

int main(void) {
	int a, b, c;

	a = 3;
	b = 2;

	c = sum(a, b);

	printf("The sum of a + b = %d + %d = %d\n", a, b, c);

	return 0;
}
```

All code in a **single** file

gcc -c demo_v1.c

gcc -o demo_v1 demo_v1.o

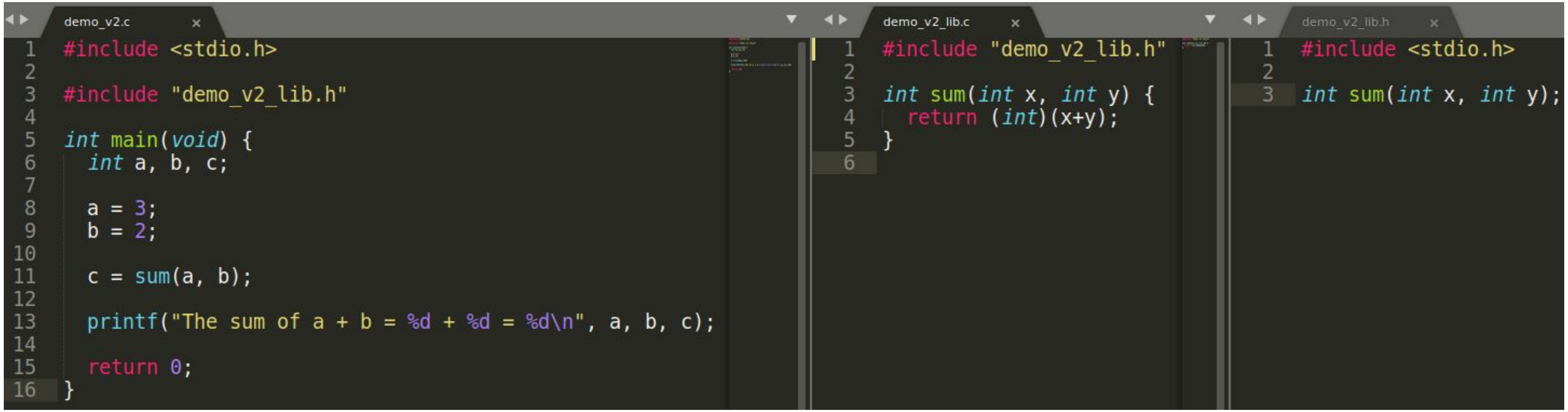
C source
object file
binary
static library



```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}

```

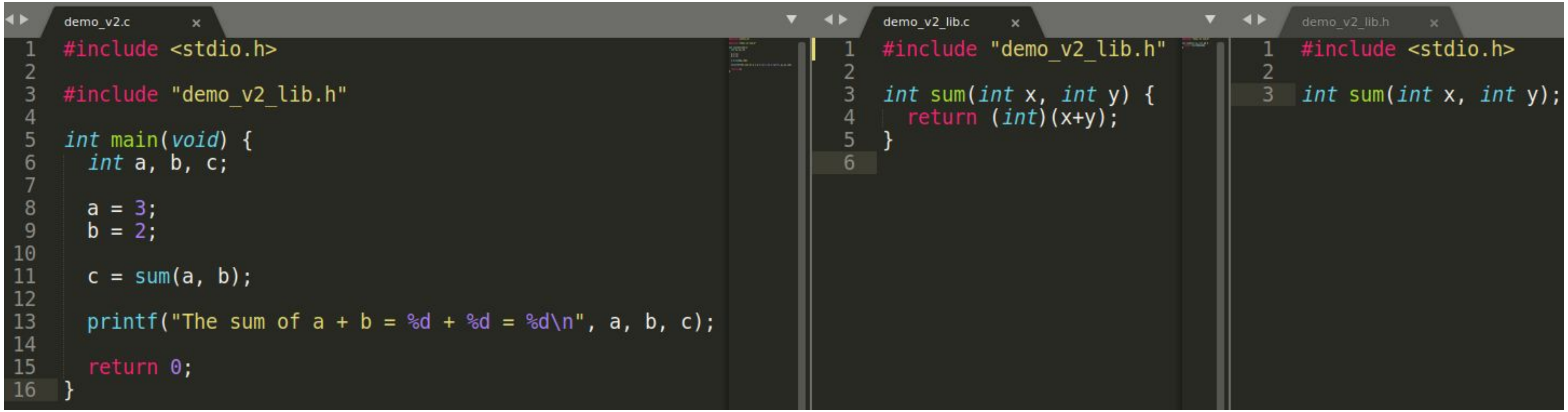
```c
#include <stdio.h>

int sum(int x, int y);
```



```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}

```

```c
#include <stdio.h>

int sum(int x, int y);
```


demo_v2.c

```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

demo_v2_lib.c

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}

```

demo_v2_lib.h

```c
#include <stdio.h>

int sum(int x, int y);
```

All code in **separate** files

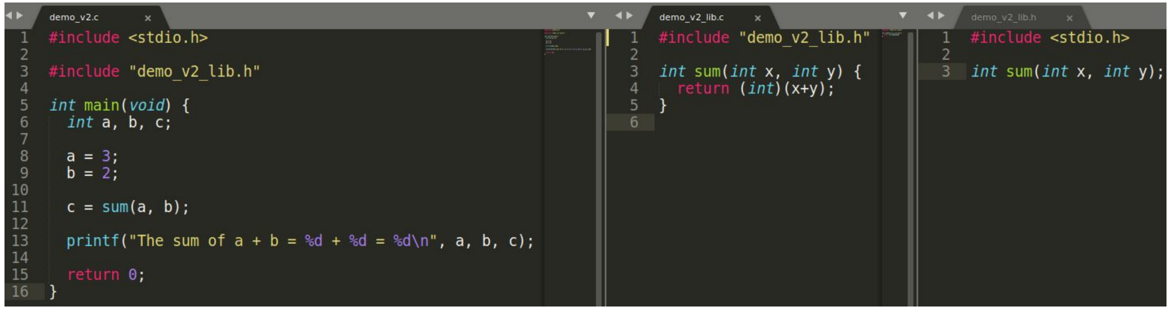
gcc -c demo_v2_lib.c

gcc -c demo_v2.c

gcc -o demo_v2 demo_v2.o demo_v2_lib.o

C source
object file
binary
static library

code is unaltered !!


demo_v2.c

```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

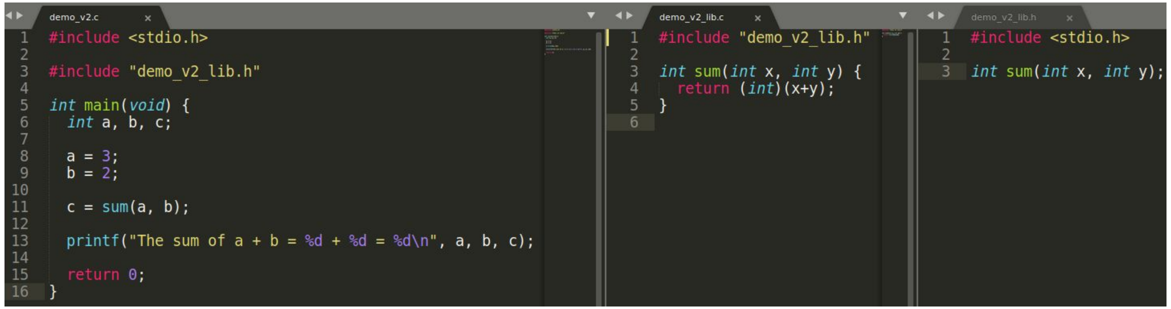
demo_v2_lib.c

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}

```

demo_v2_lib.h

```c
#include <stdio.h>

int sum(int x, int y);
```

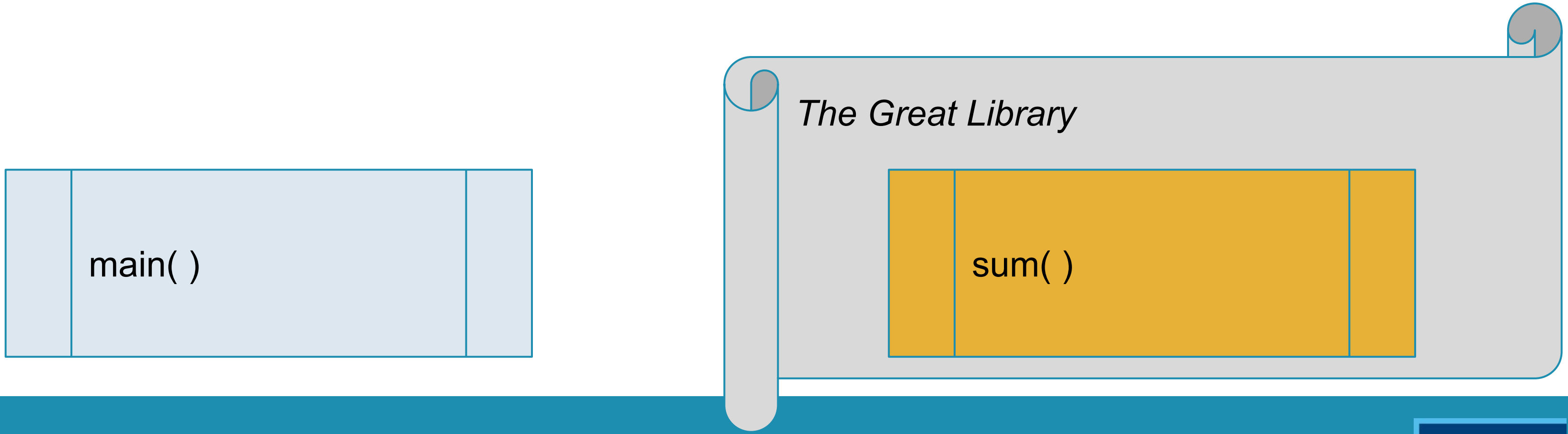
KU LEUVEN



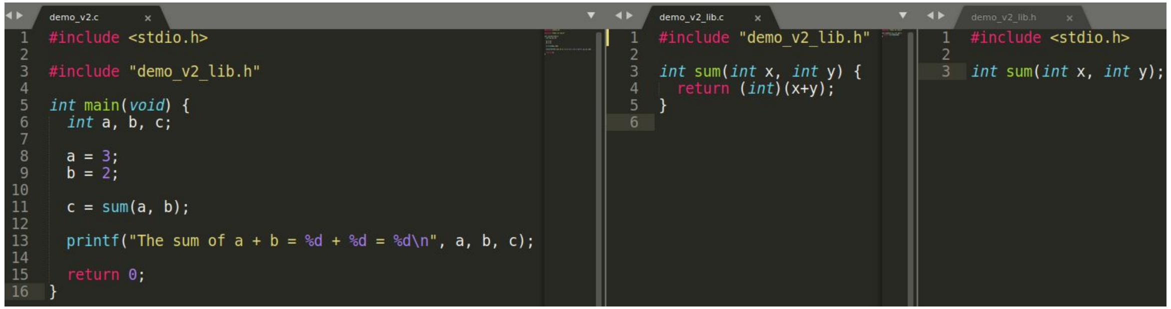
```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}
```

```c
#include <stdio.h>

int sum(int x, int y);
```



code is unaltered !!


demo_v2.c

```c
#include <stdio.h>

#include "demo_v2_lib.h"

int main(void) {
  int a, b, c;

  a = 3;
  b = 2;

  c = sum(a, b);

  printf("The sum of a + b = %d + %d = %d\n", a, b, c);

  return 0;
}
```

demo_v2_lib.c

```c
#include "demo_v2_lib.h"

int sum(int x, int y) {
  return (int)(x+y);
}

```

demo_v2_lib.h

```c
#include <stdio.h>

int sum(int x, int y);
```

```
gcc -c demo_v2_lib.c
ar -rcs libdemo_v2.a demo_v2_lib.o
gcc -c demo_v3.c
gcc -o demo_v3 demo_v3.o -L. -ldemo_v2
```

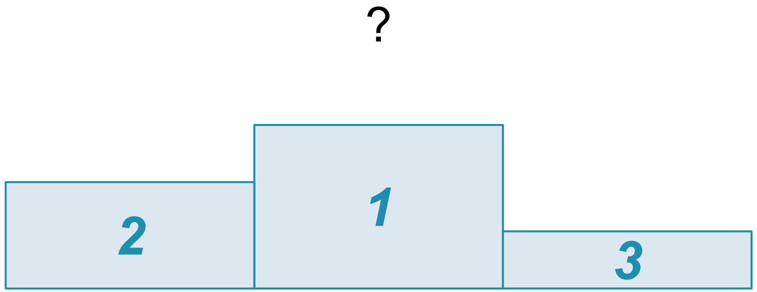
C source
object file
binary
static library

- ES&S
- Introduction
- Library exercise
- Experiment
- Your turn !

Setting

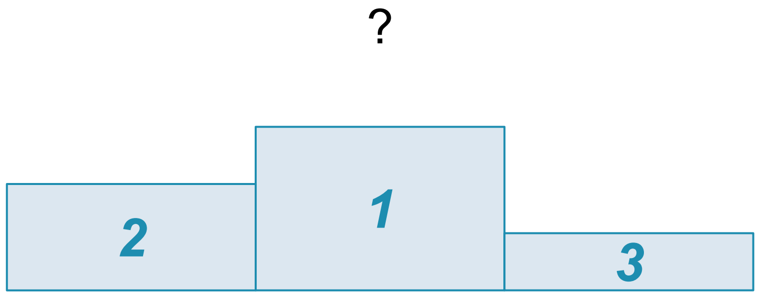
- Ongoing competition: https://csrc.nist.gov/projects/lightweight-cryptography/

- 10 finalists



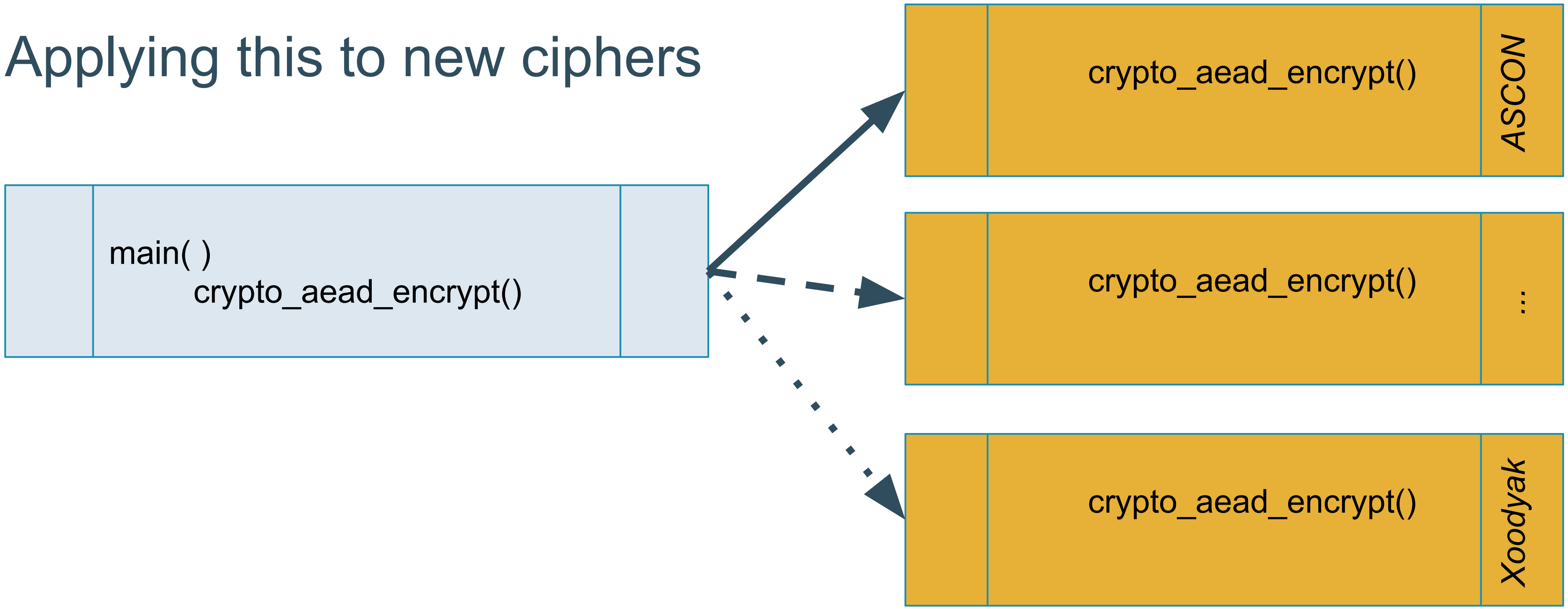
Setting

- speed
- file size
- optimal input size
- ...



A little help

Applying this to new ciphers



- ES&S
- Introduction
- Library exercise
- Experiment
- Your turn !

Labsetup

- WiFi: eduroam / campusroam
- Server IP address: 10.4.14.1

  only accessible in Building B or TCN

- User profiles: mol_user*n* with *n* {1, 2, 3, 4, 5},

  e.g.: user: “mol_user2”, pw: “mol_user2”

Slides: https://github.com/jvliegen/mol_esands