CSCI251/CSCI851 Autumn-2020 Advanced Programming (**LT2**)

Lecture Tutorial 2

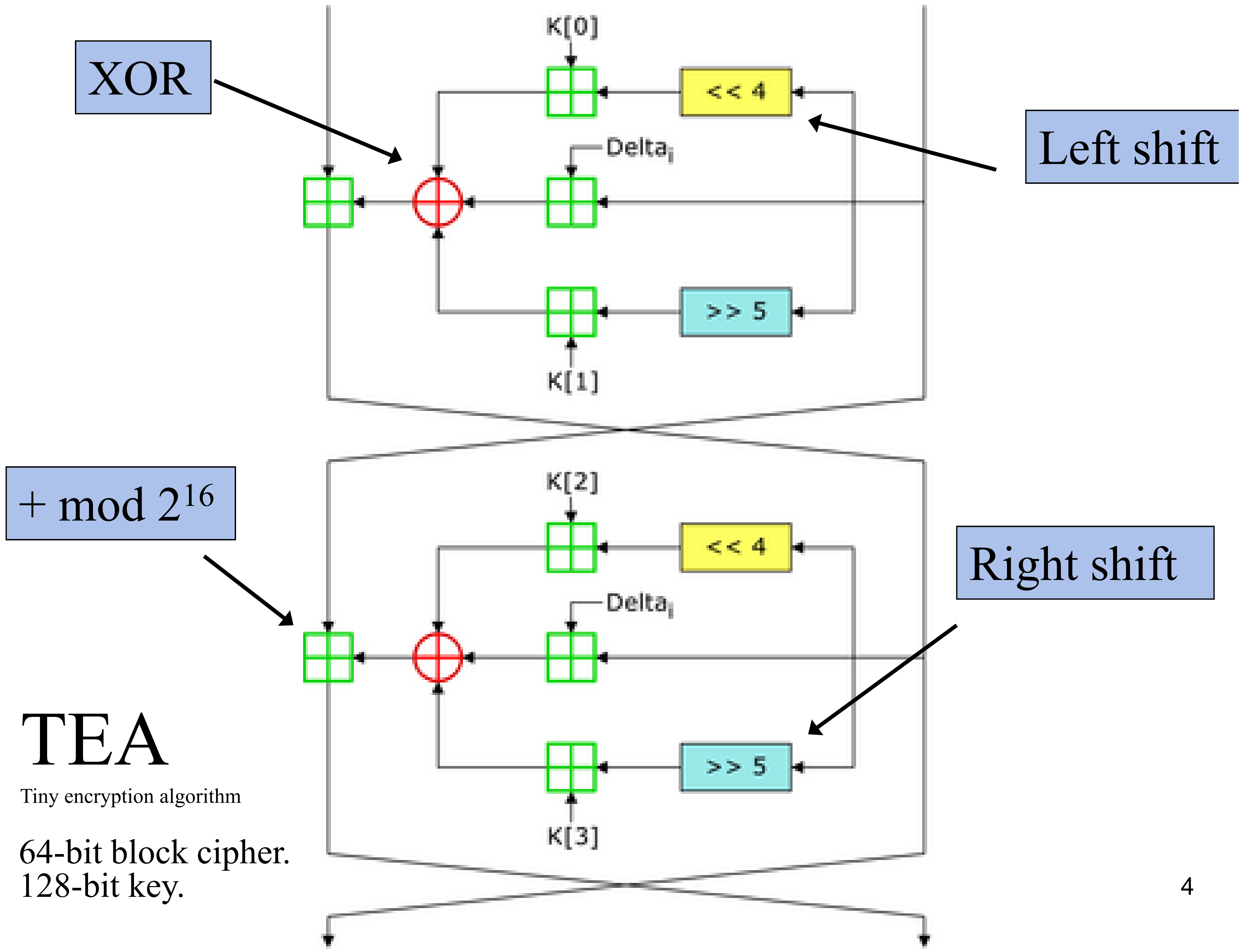
Outline

- From the lab:
- Pass by reference, pass by value…
- Default arguments.
- Proceeding procedurally.

From the lab:

```
cout << (value << 2) << endl;
```

- The first << and the second << differ …
  - << : Insertion, left shift.
  - >> : Extraction, right shift.
- The bracketed part gets calculated first and it’s recognised that the arguments to << or >> are different.
- They are shifting a binary representation to the left or to the right, thus the names.
  - These multiply or divide by 2.



TEA

Tiny encryption algorithm

64-bit block cipher.
128-bit key.

TEA Encryption

```
void encrypt(unsigned long* v, unsigned long* k) {
   unsigned long v0=v[0], v1=v[1], sum=0, i;    // setup
   unsigned long delta=0x9e3779b9;      // key schedule constant
   unsigned long k0=k[0], k1=k[1], k2=k[2], k3=k[3];  // cache key
   for (i=0; i < 32; i++) {                         // start round
        sum += delta;
        v0 += (v1<<4)+k0 ^ v1+sum ^ (v1>>5)+k1;
        v1 += (v0<<4)+k2 ^ v0+sum ^ (v0>>5)+k3;  // end round
   }
   v[0]=v0; v[1]=v1;
}
```

The operation ^ represents bitwise XOR.
The values being unsigned here is critical to the modular addition working correctly.

Pass by reference, pass by value

- C++ has 2 ways to pass variables to functions:
  - Pass by value, to be used when the function doesn't need to change the value of the arguments given to it.

```
return_type function_Name (type var1, type var2, …);
int get_larger (int A , int B);
```

  - Pass by reference, to be used when the function may change the arguments.

```
return_type function_Name (type &var1, type &var2, 
…);
int sort (int &A , int &B);
```

Default arguments

- We can have a look at an example of this.
- Add an extra function, with a default argument, to another program.

Proceeding procedurally

- What does an actual procedural program look like?
  - What is the process?
    - `main()`
    - Stubs.
    - Stubs vs. prototypes.

The task at hand …

- Read in a collection of numbers and sort them in increasing order before displaying them.
- The parts based on that:
  - Reading the data → a collection of numbers.
  - Sorting the data.
  - Displaying the data.
- We can write procedures/functions for each of these.