

# Mauborgne Cipher - a modern version of OTP

**Use a random stream** as encryption key

➡ Defeats the know-plaintext attack

**Problem** : Key-reused attack (a.k.a two-time pad)

$$C_1 = k \oplus m_1$$

$$C_2 = k \oplus m_2$$

$$\begin{aligned}\text{so } C_1 \oplus C_2 &= (k \oplus m_1) \oplus (k \oplus m_2) \\ &= (m_1 \oplus m_2) \oplus 0 \\ &= (m_1 \oplus m_2)\end{aligned}$$

|                  |
|------------------|
| $x \oplus x = 0$ |
| $x \oplus 0 = x$ |

# Random Number Generator

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

## True Random Number Generator

➔ No, because we want to be able to encrypt and decrypt

## Pseudo-Random Generator

➔ Stretch a fixed-size seed to obtain an unbounded random sequence

