CTR-Mode Encryption

Helger Lipmaa

Helsinki University of Technology (Finland)
University of Tartu (Estonia)

Phillip Rogaway

University of California - Davis (USA)
Chiang Mai University (Thailand)

David Wagner

University of California - Berkeley (USA)

What is CTR Mode?

* The simplest correct way to encrypt using a block cipher

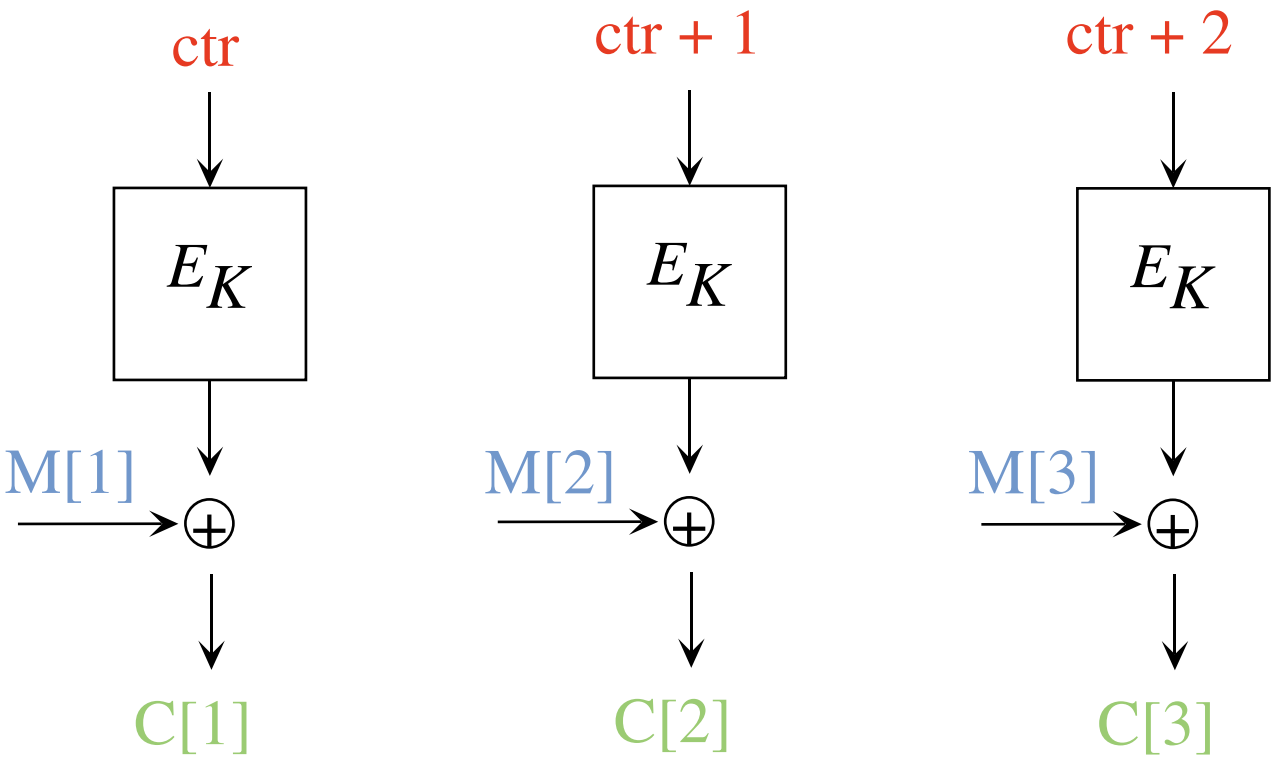
* An old mode, dating to DH79, but omitted from earlier FIPS

* A Vernam cipher (like a one-time pad), but no state is maintained by the sender

Why the renewed interest?

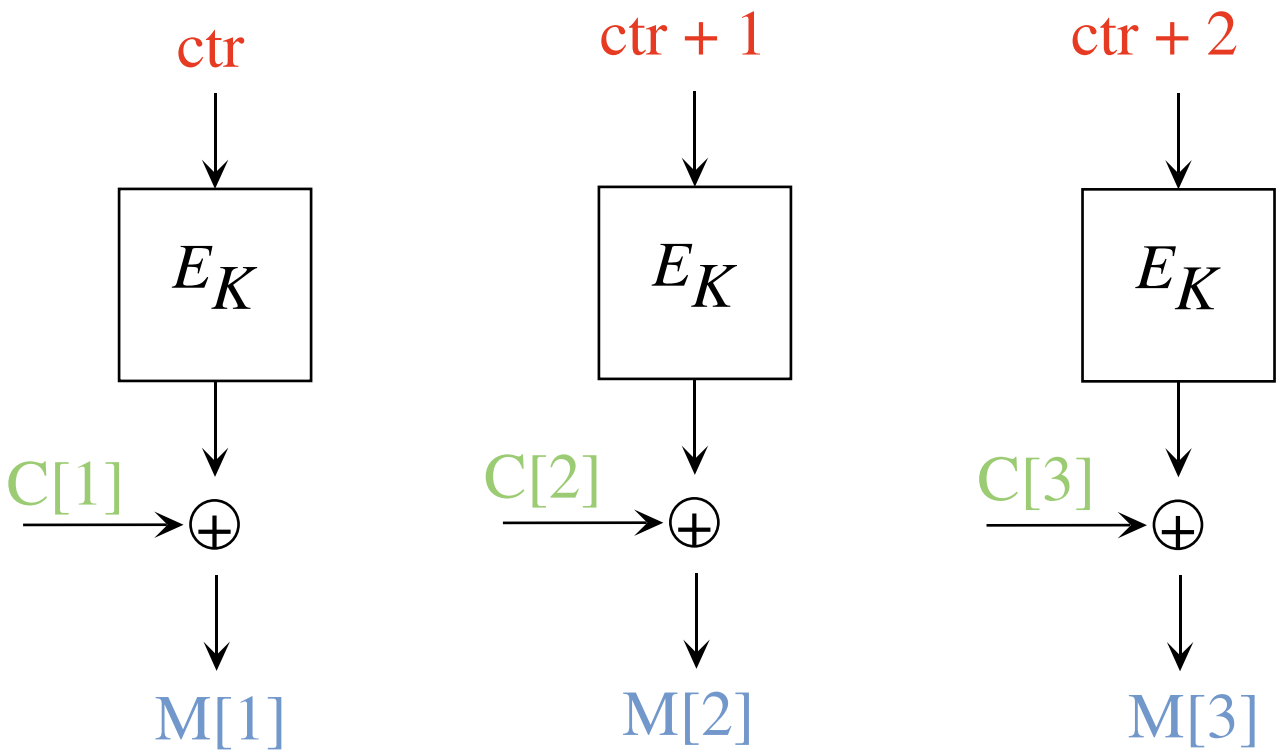
* Because CTR mode is fully parallelizable, making it much more efficient, in many contemporary usage scenarios, than modes like CBC.

CTR Mode Encryption



The ciphertext is C[1] C[2] C[3] and something adequate to recover ctr

CTR Mode Decryption



The plaintext is M = M[1] M[2] M[3]

Where does the ctr come from?

* It is supplied on the encrypting side (like the IV in CBC mode)

* It is **crucial** that no ctr+i value be repeated - repeating such a value is like reusing a one-time pad.

* Recommended way of making ctr :

$$\text{ctr} = \text{nonce} \;\|\; 0000 \cdots 0000$$

..64 bits .. *... 64 zero bits ...*

Advantages

* Faster SW speed on modern processors (Itanium, Alpha, AltiVec, etc.)
* Essentially unlimited HW speed
* Provably secure (Same bounds as CBC MAC, same assumption [BDJR])
* Random access to the "middle" of the ciphertext
* Preprocessing possible
* Arbitrary message lengths
* No need to implement E^{-1}
* Completely patent-free

Complaint	**Answer**
No integrity	Right. Just like all the other conventional encryption modes. For integrity, use a MAC or an authenticated-encryption mode.
No error propagation	So what.
Sender needs state or $	Right. True of any secure enc scheme
Sensitive to usage errors	Some validity. Be clear : *do not reuse a ctr value* ! Counter/nonce distinction helps
Quad sec bound	Like other modes; n=128 makes OK
Interaction with weak ciphers	Use with strong block cipher