

Encrypted Storage - Challenges and Methods

James Hughes

Storage Technology Corporation

Agenda

- Vulnerabilities of Storage
 - What and how much
 - Destruction
- Encryption
 - Where to do it
 - Software
 - Hardware
 - Existing Encryption Algorithms
 - Modes
- Conclusion
 - standards
 - The future

Vulnerabilities of Storage

- What and how much
- Residuals
- Destruction

Storage is different

- Communications
 - Build a key, use a key, destroy a key
 - Loss of a key
 - get a new (unrelated) key
- Storage
 - Build a key, use a key, keep the key
 - Loss of a key
 - Loss of data (or else)
- Communications assume storage channel
 - Storage is a storage channel
 - without the effort

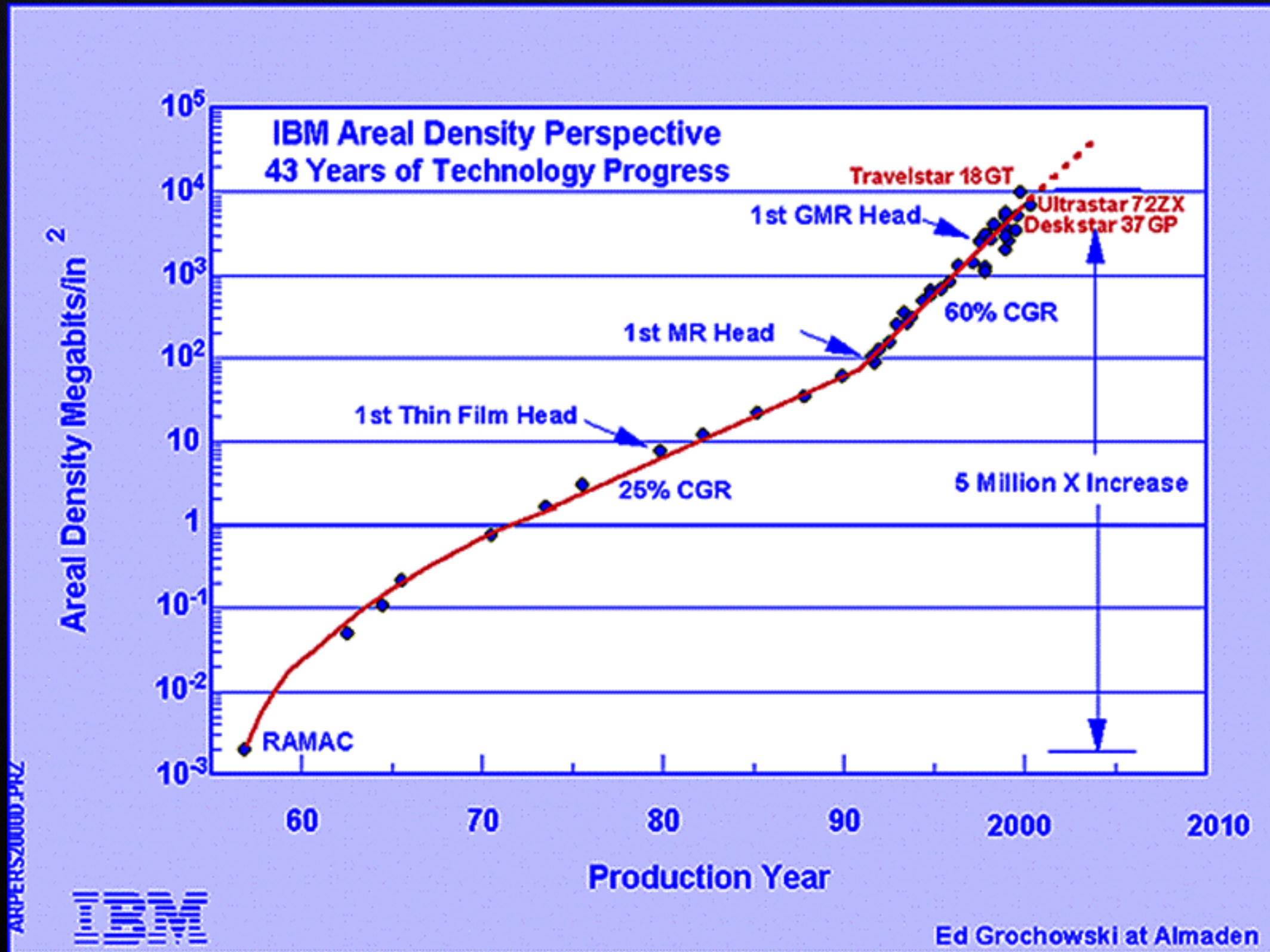
What Do You Store?

- What you have
 - Done
 - Valued
 - Intend

SneakerNet

- What will be the most cost effective method of data transfer
 - “Over the last 40 years telecom prices have fallen much more slowly than any other information technology.” Jim Gray (Microsoft)
- The economics of data movement over communications lines is getting less viable
- The economics of data movement using physical means is becoming more important, not less

How Much Storage



Destruction of Data in a Hurry

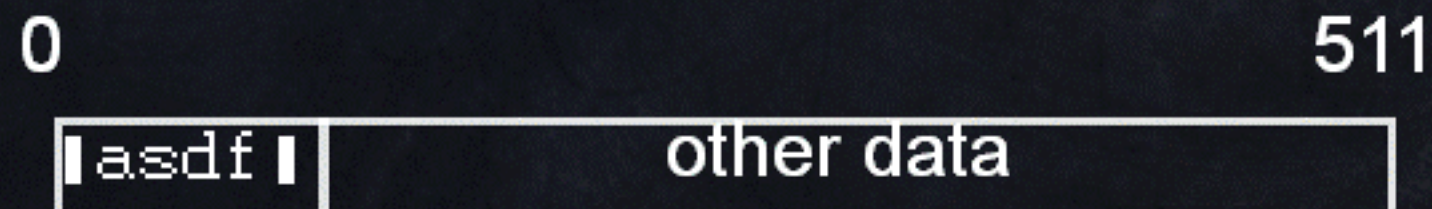
- ① Deleting the file
- ① Over writing the data
- ① Shoot the drive
- ① Security Erase
- ① De-Gaussing
- ① Melting

Deleted Files

- Disks have many 512 byte blocks
- Free chain
- “Metadata”
 - Name
 - start, length
 - Create, modify and access times
- Delete a file
 - erase metadata
 - put storage back on free chain
 - Does not remove the information
- Raw read of the disk will provide the data

Buffer Remnants

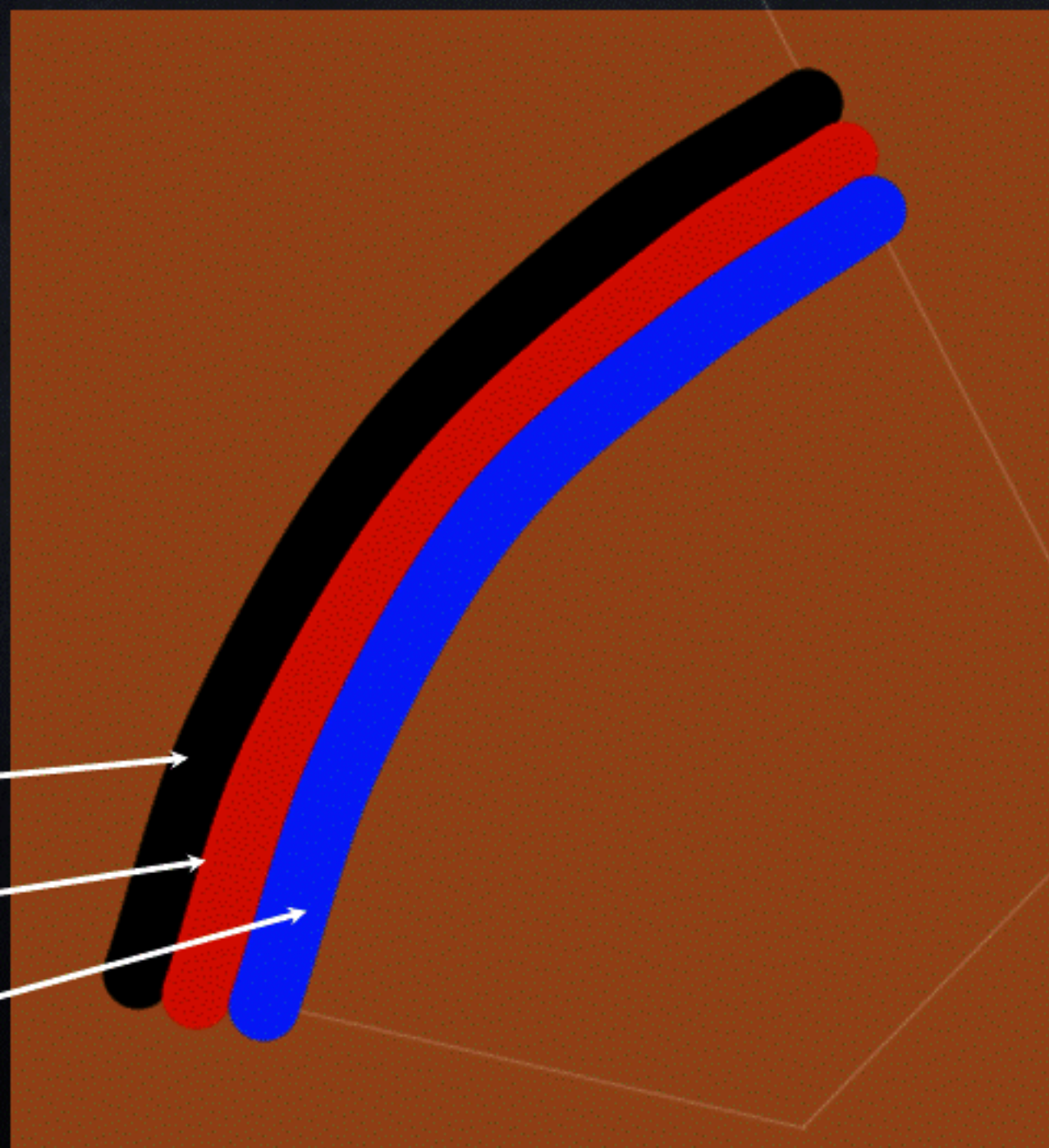
- Write data of integer number of bytes
 - `write(out, "asdf", 4);`
- Write data of integer number of sectors
 - 512 bytes
- Does not clear out to end of sector



Over Written

- Off-track
- Phase noise

Off Track Remnants

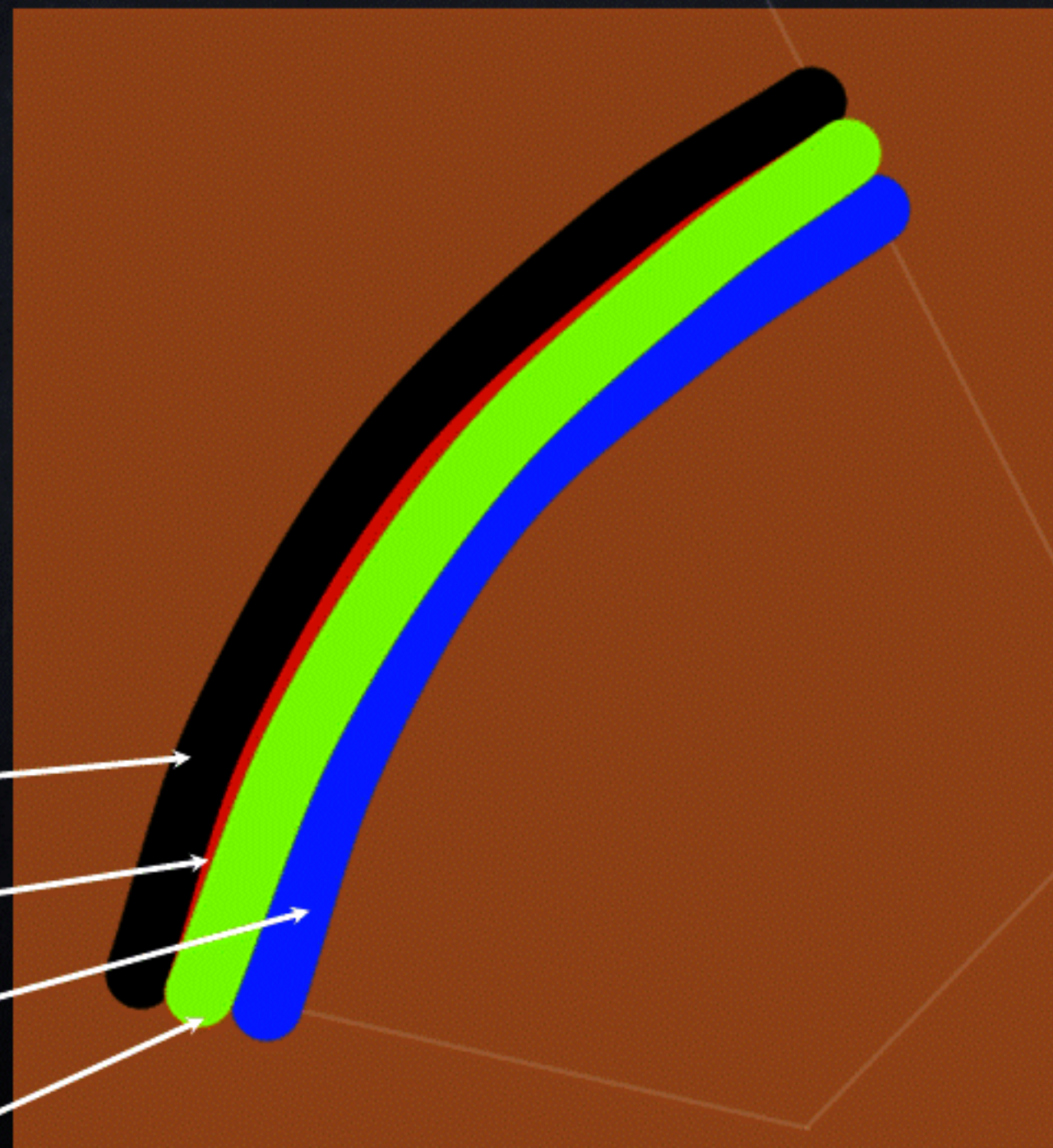


Track 1

Track 2

Track 3

Off Track Remnants



Track 1



Track 2



Track 3



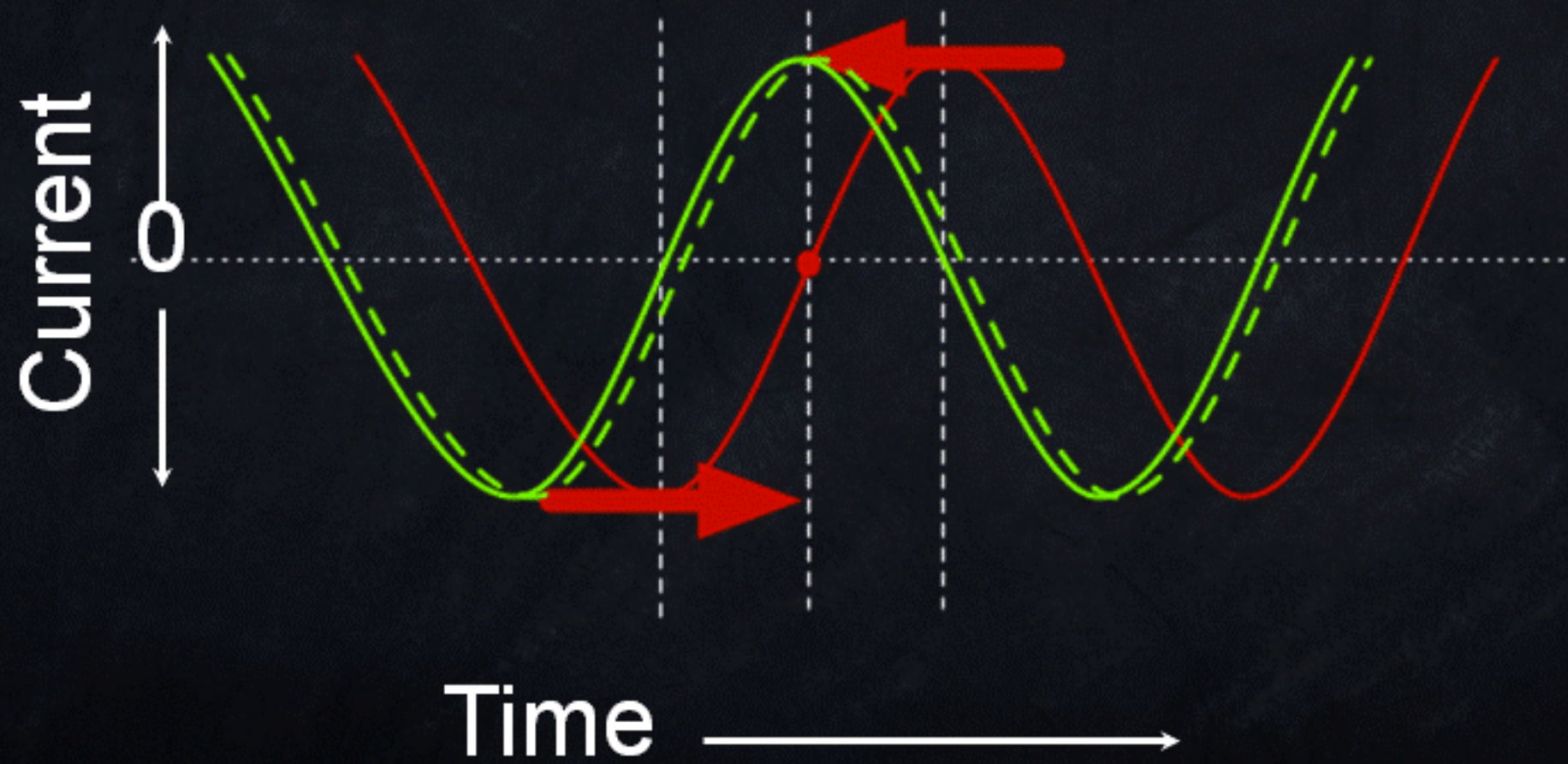
Overwrite



Phase Noise

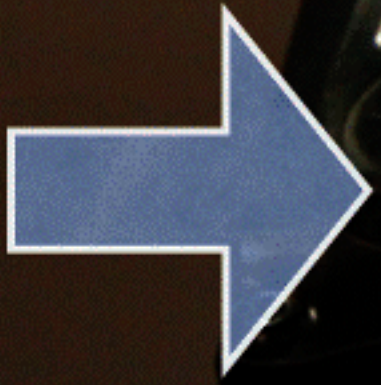
- Current Disk Drives
 - Tracks are no longer erased
 - Written over old data
 - “Noise” budget
- Conflicting Science
 - 5% Peter Gutmann
 - None. Gordon Hughes
- Media Dependent

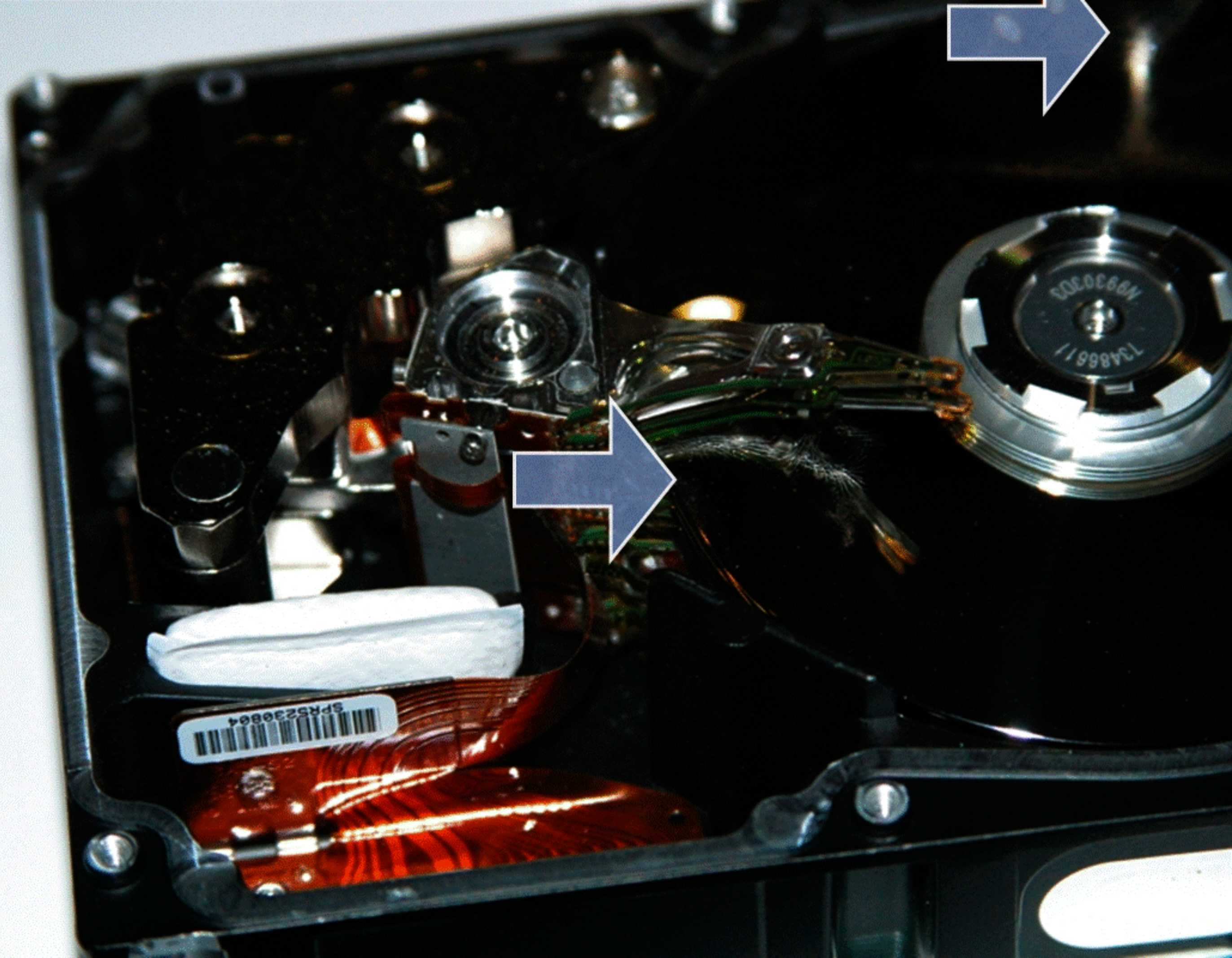
Phase Noise



Shoot the disk?

- Renders the disk “unspinnable”



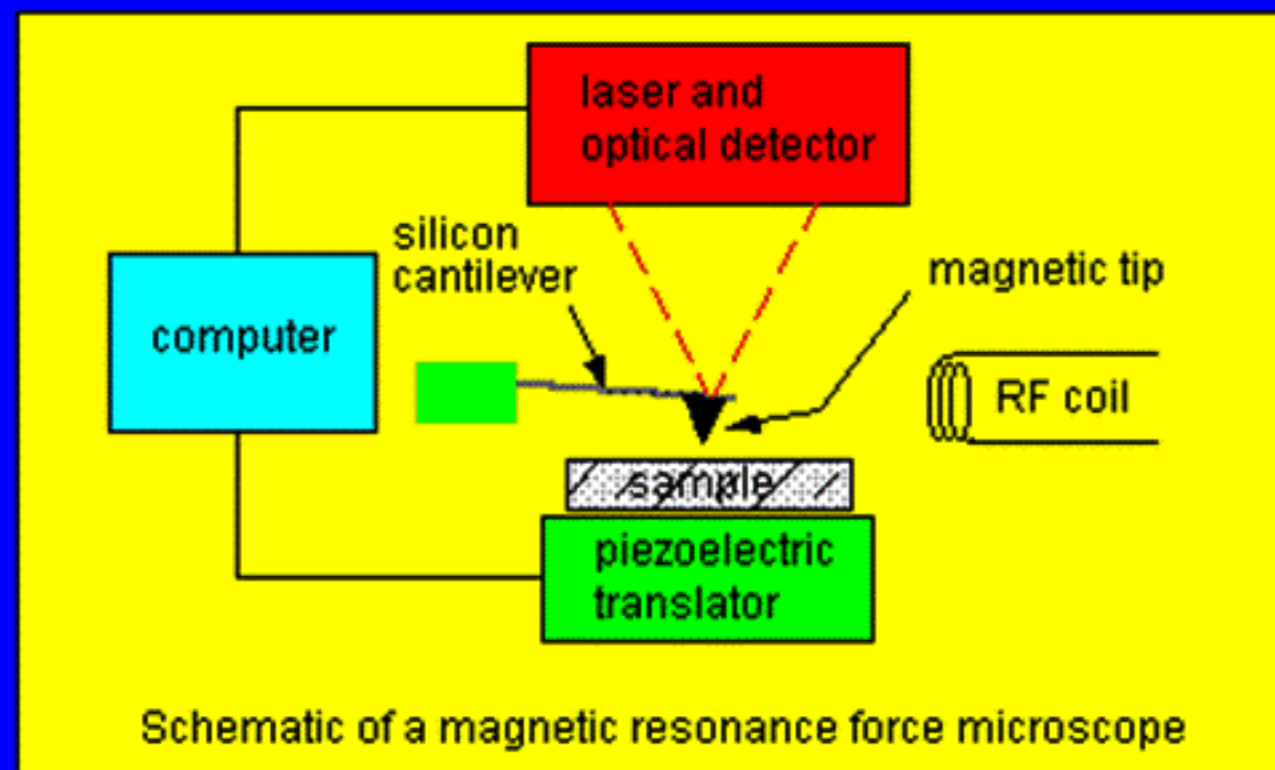


SPR52230804

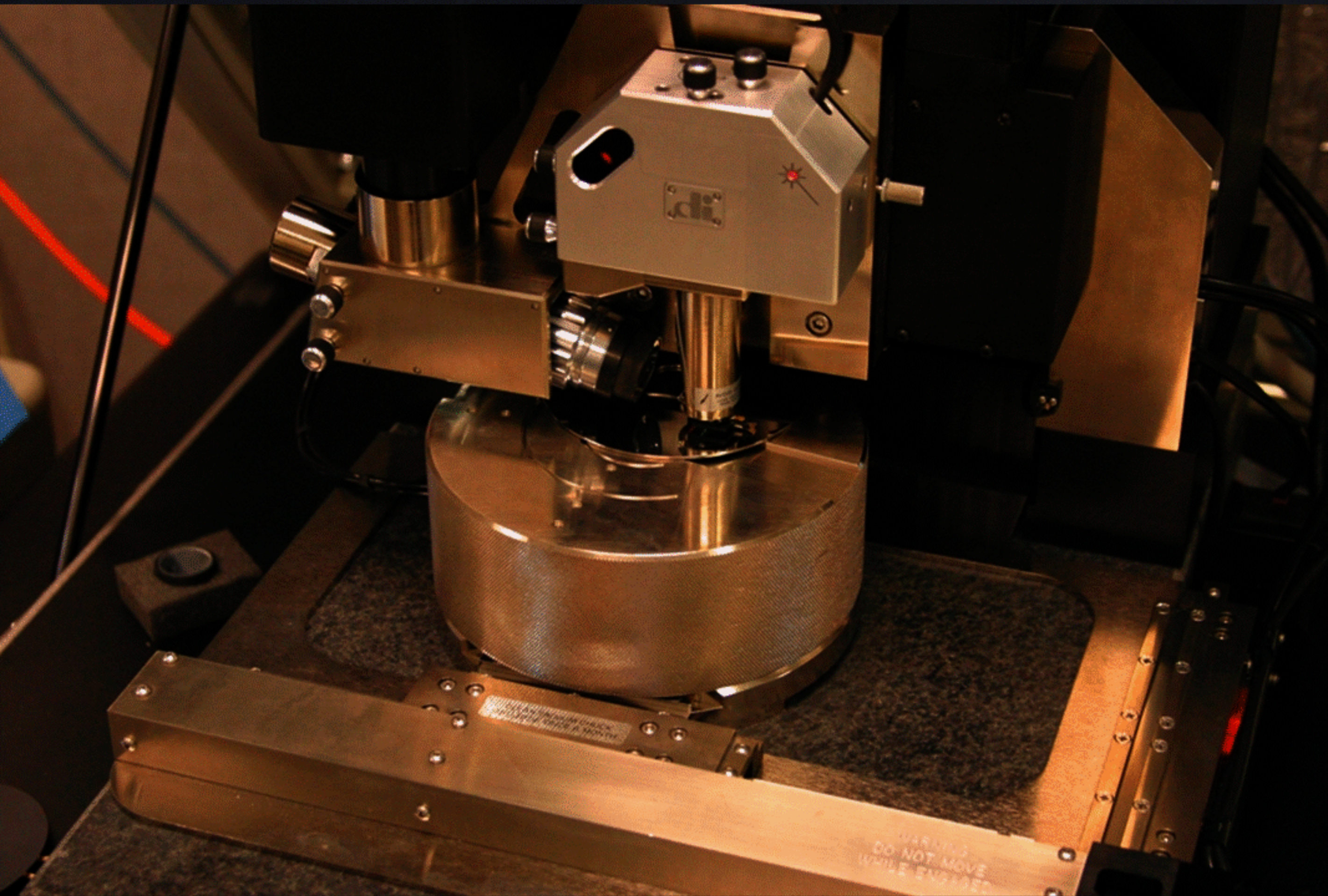
7348661
M930303

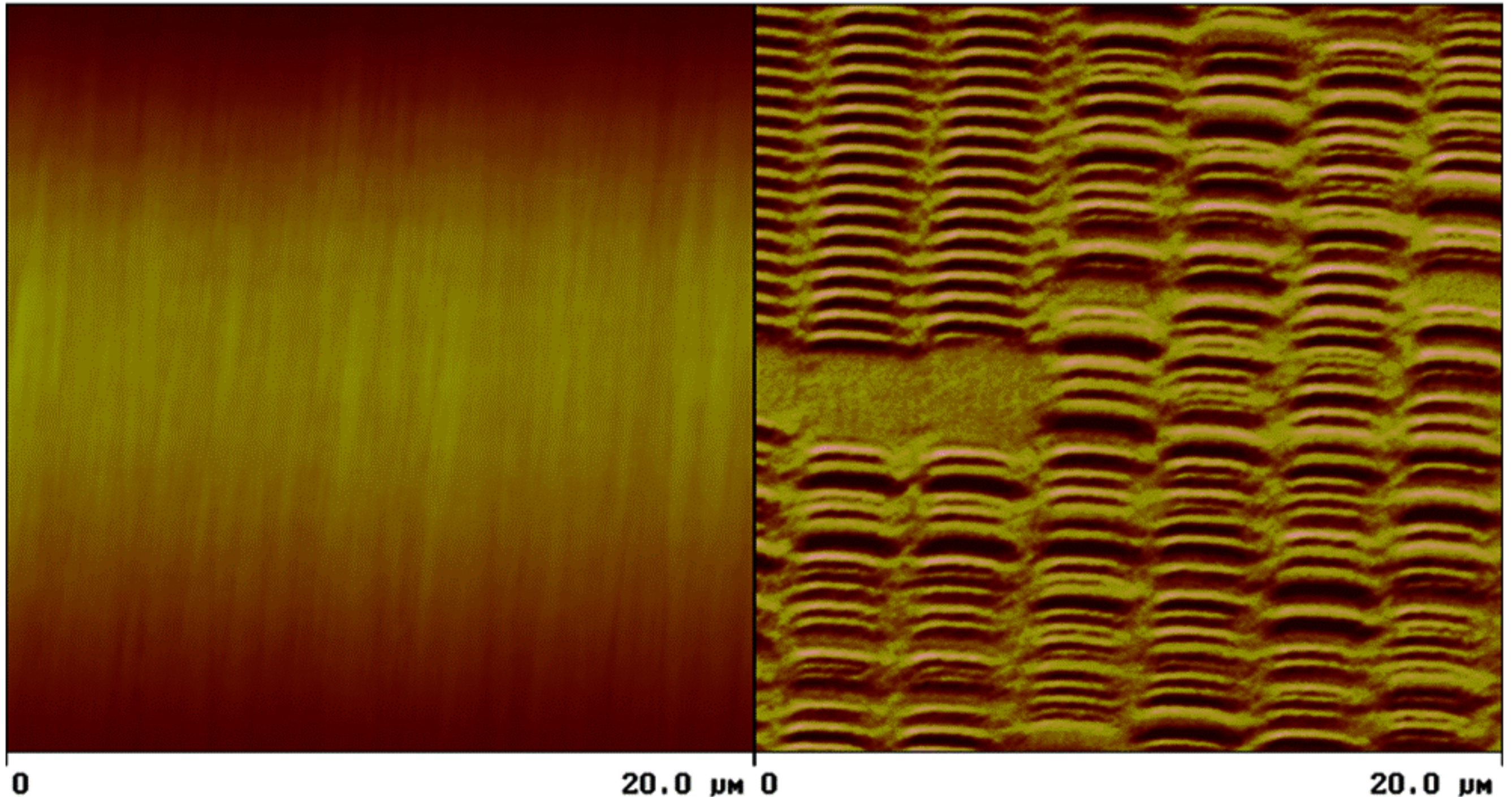


Magnetic Force Microscope



<http://www.boulder.nist.gov/magtech/mrfmsil.gif>





0 20.0 μm 0

Data type
Z range

Height
100 nm

Data type
Z range

Frequency
5.00 Hz

20.0 μm

Degaussing Machines



Destruction of Data in a hurry

- ① ~~Deleting the file~~
- ① ~~Over writing the data~~
- ① ~~Shoot the drive~~
- ① ~~Security Erase~~
- ① ~~De-Gaussing~~
- ① ~~Melting~~
- ① Encryption?
 - ① Delete the key, delete the data

Encryption

- Where to do it
 - Software
 - File Encryption (e.g. PGP, GPG, etc.),
 - Volume Encryption,
 - Hardware
- Existing Encryption Algorithms
 - Modes

File Encryption

- PGP, GPG
- Encrypt creates a container
 - $k_1 = \text{random}$
 - $F = \text{IDEA}_{k_1}[\text{file}]$
 - $H = \text{hash}[k_1, \text{file}]$
 - $K = \text{RSA}_{k_e}[k_1]$
 - $(F | K | H)$
- Decrypt removed the container
 - $k_1 = \text{RSA}_{k_d}[K]$
 - $\text{file} = \text{IDEA}_{k_1}[F]$
 - $H = ? = \text{hash}[k_1, \text{file}]$

File Encryption Vulnerabilities

- Other files
 - Temp, print, source, etc.
 - Deleted?
 - Where do you draw the line
- Swap space
- Metadata
 - file names
 - modification dates and times
- Key Management
 - Random numbers
- File encryption works best for email

Volume Encryption

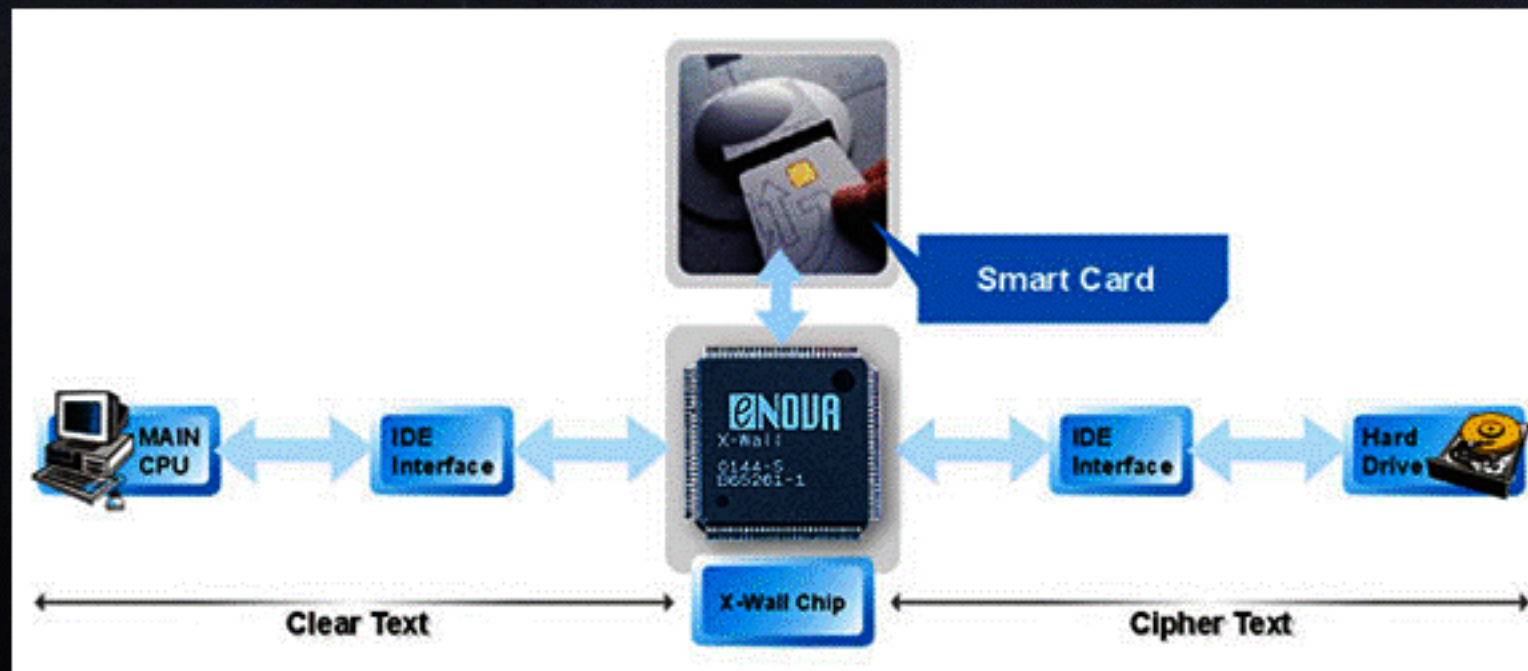
- Software
 - PGP Disk, Loopdriver, etc.
 - Key Management
 - requires clear boot/OS
- Hardware

Open Source Software

- Allows an escrow of the source
 - Look back and understand
 - Does not replace
 - testing
 - code reviews
- Linux
 - Loop driver (old)
 - dmccrypt (new)

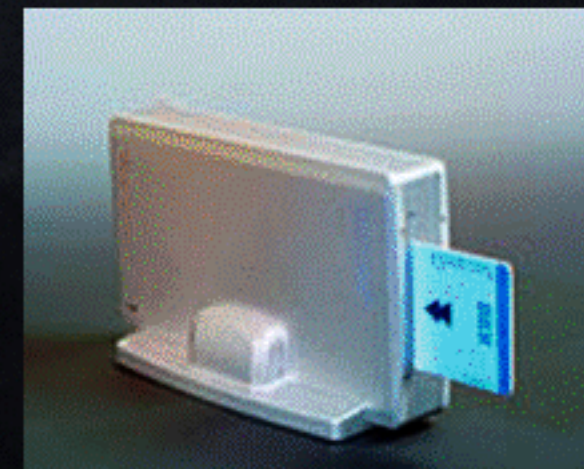
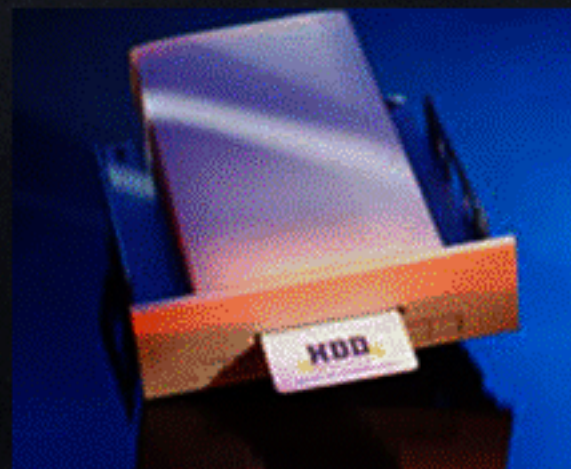
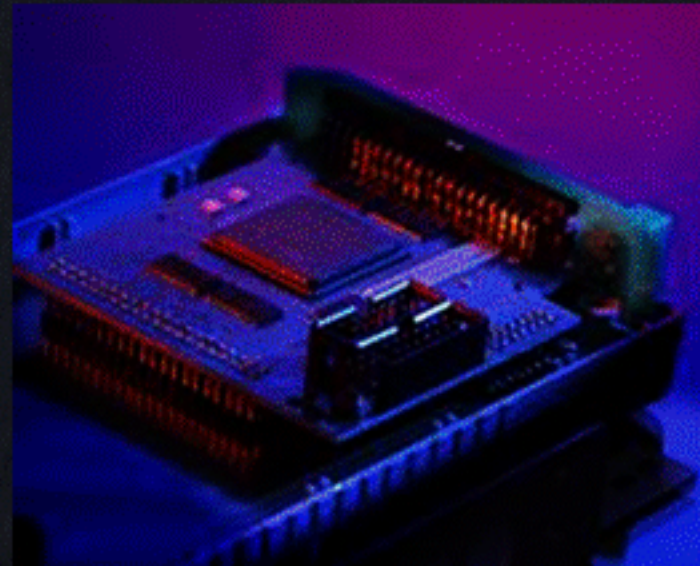
Hardware

- Encrypts everything
 - 40, 56 bit DES
 - 112-168 bit TDES
 - CBC?
 - “Pre keyed”?




Hardware

- ① <http://www.hdd.no/>
- ① Nothing in the clear
 - ① AES
 - ① CBC



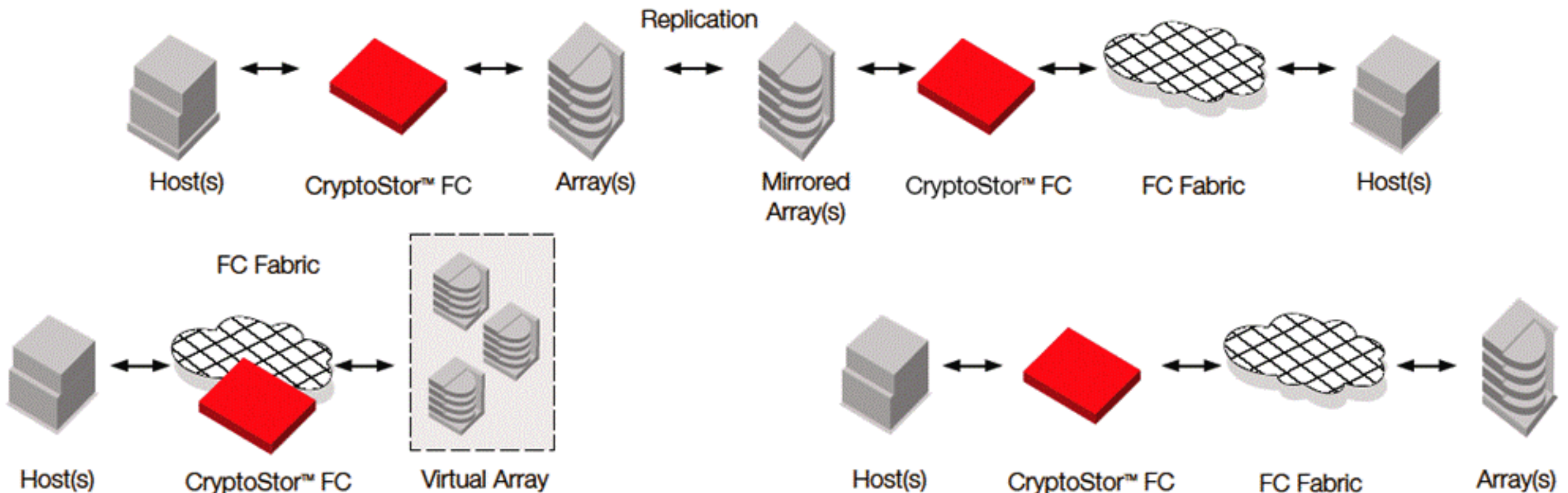
SAN Based Encryption

 Neoscale

CryptoStor™ FC High-Performance Storage Security Appliance

Enterprise Class Data Security

- Storage Firewall access control
- Primary storage encryption
- True wire-speed performance
- Fully transparent protection
- Centralized policy management
- Scalable deployment
- Strong security standards



Terminology

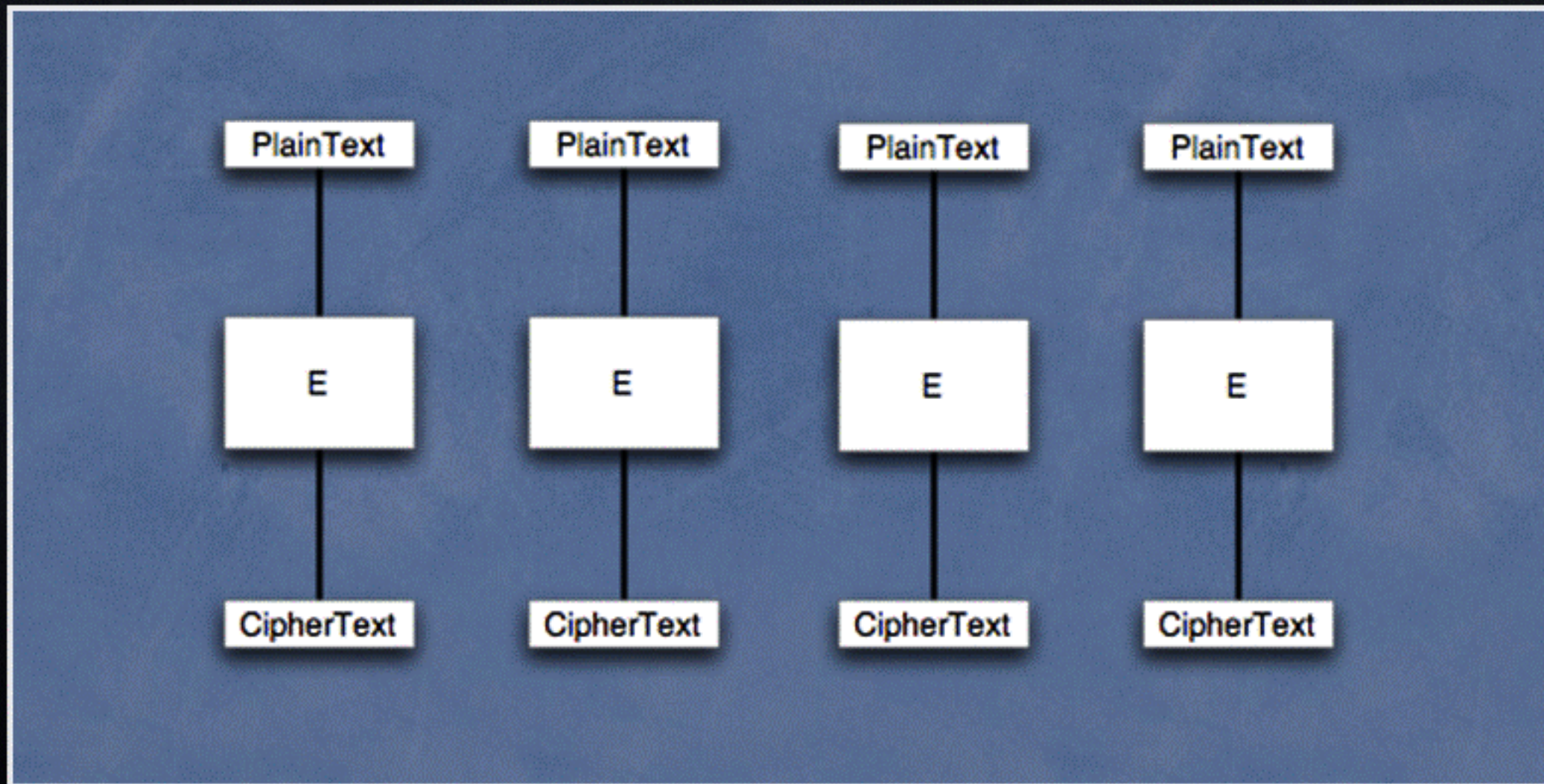
- Sector
 - Disk hardware unit of transfer (512 bytes)
- Block
 - Cipher unit size (8 or 16 bytes)

Encryption Modes

- Existing solutions use Cipher Block Chaining
- A way of using a block cipher
 - Extension of the cipher
- Electronic Code Book
 - $C = E[P]$
 - Whenever P is encrypted C is produced
 - Permutation of the group
 - Dictionary attack
 - If $C_1 = C_2$ then $P_1 = P_2$
- Assume a perfect cipher
 - Attacks performed without reversing the key

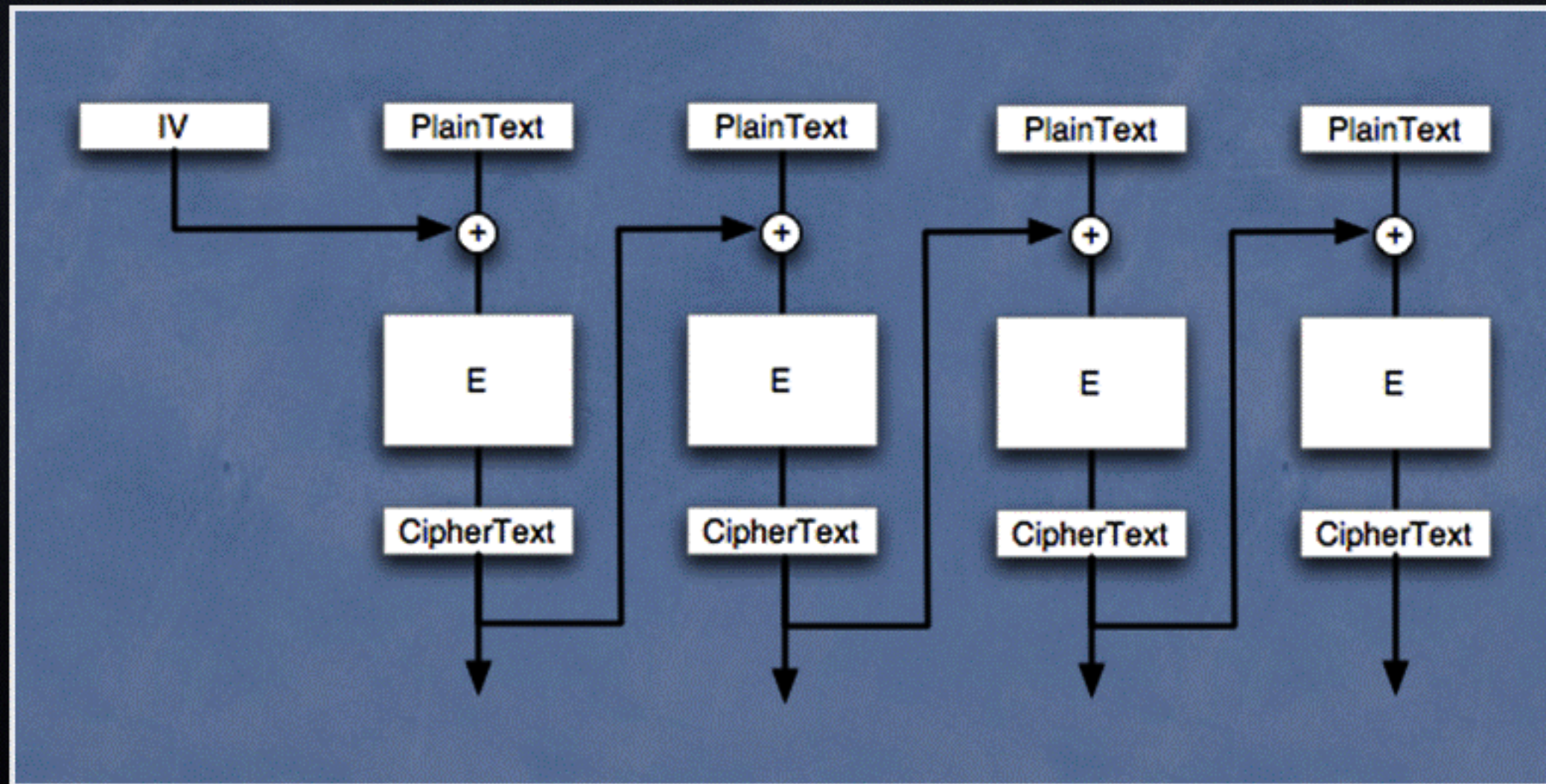
Modes	Leaks	Watermark	Malleable	Movable	Replay-Block	Replay-sector
ECB	Yes	Yes	No	Yes	Yes	Yes
Plain IV	Yes	Yes	Yes	Yes	Yes	Yes
Enc IV	Yes	No	Yes	Yes	Yes	Yes
Plumb-IV1	Yes	?	Yes	Yes	Yes	Yes
LRW	No	No	No	No	Yes	Yes
CMC	No	No	No	No	No	Yes
EME	No	No	No	No	No	Yes

ECB Mode



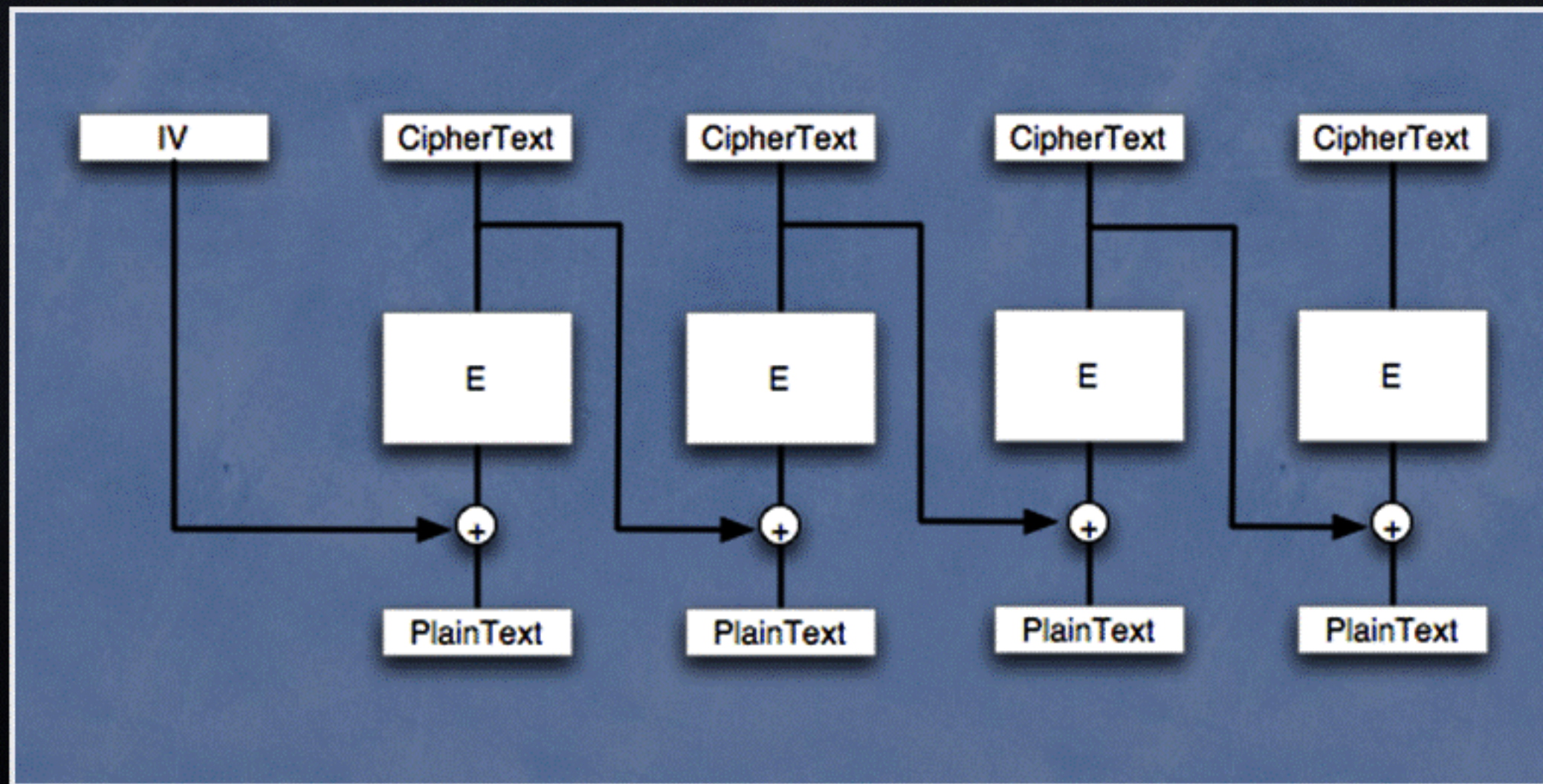
Cipher Block Chaining Mode

- Encryption



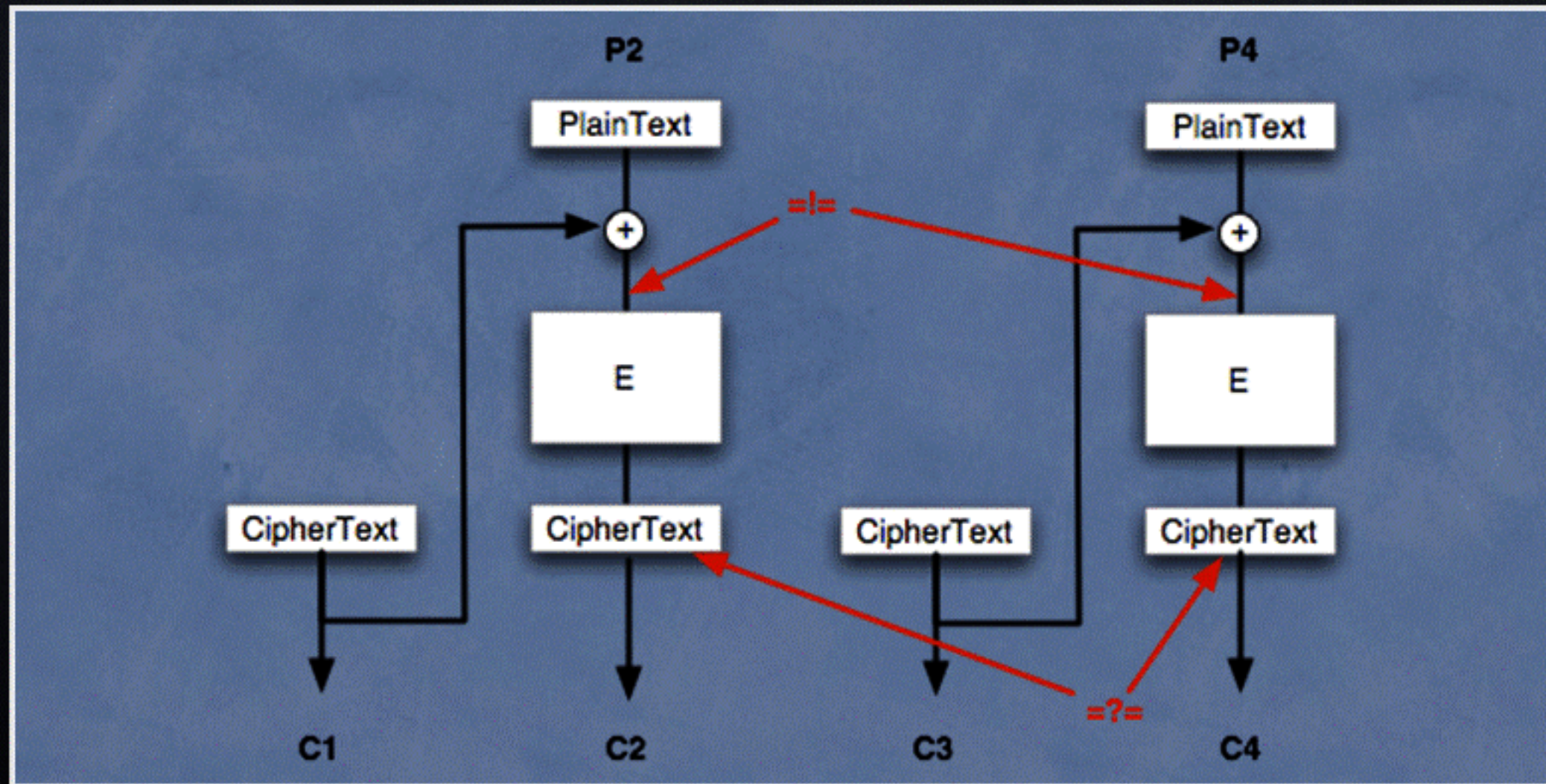
Cipher Block Chaining Mode

- Decryption



CBC Leakage

- Applies to Plain IV, Enc IV and Plumb IV



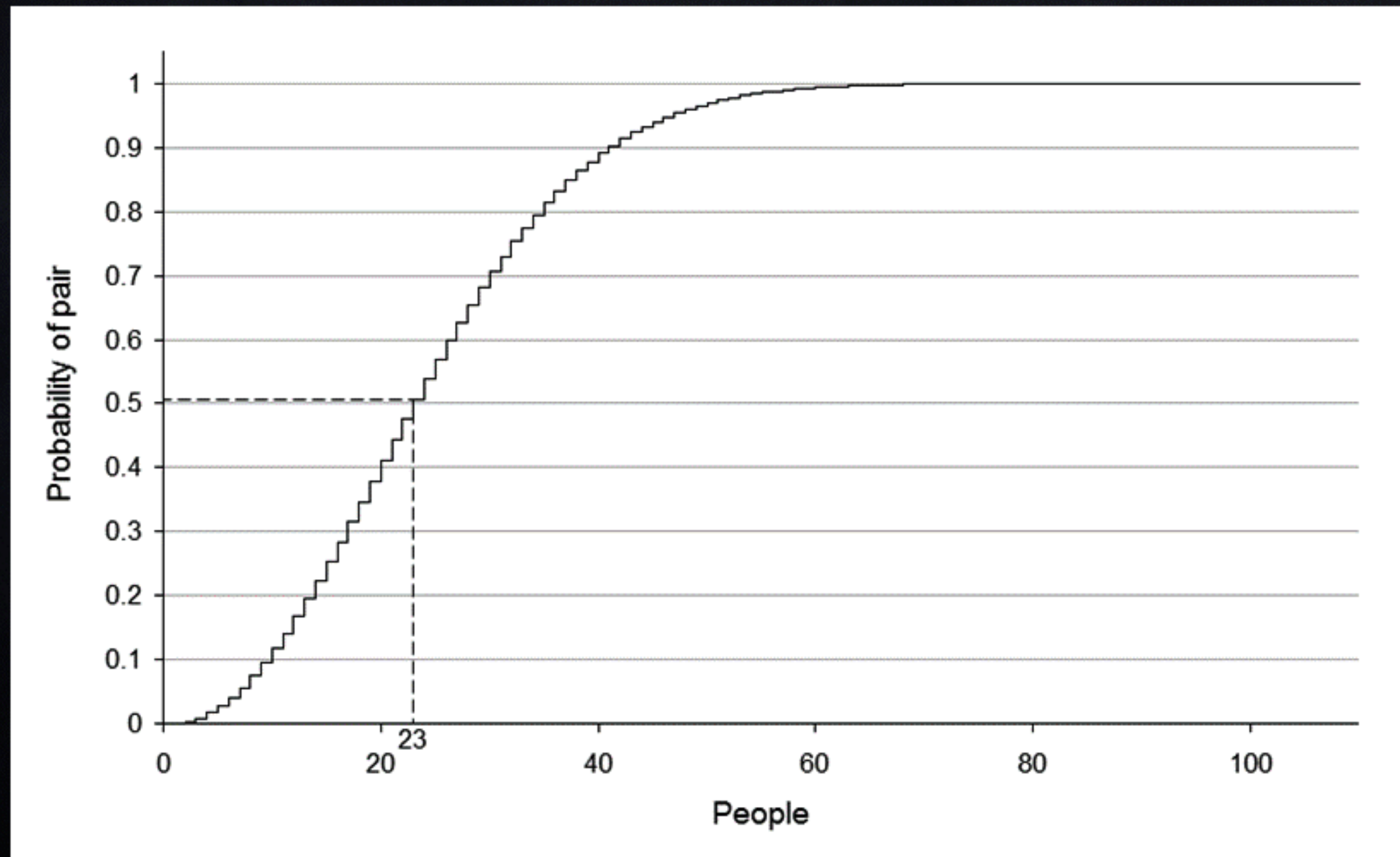
CBC Leakage

- Any Chance ciphertext blocks equal
 - $C_2 = C_4 = E[C_1 \oplus P_2] = E[C_3 \oplus P_4]$
 - $x = E^{-1}[C_2]$
 - $C_1 \oplus P_2 = x$
 - $C_3 \oplus P_4 = x$
 - $C_1 \oplus P_2 = C_3 \oplus P_4$
- We know the XOR of the plaintext
 - $C_1 \oplus C_3 = P_2 \oplus P_4$

Leakage

- Occurs after $2^{b/2}$
 - Birthday Paradox
 - 64 bit block
 - 2^{32} , 34 Gigabytes
 - DES, TDES, Etc.
 - 128 bit block
 - 2^{64} , 295 Exabytes
 - AES

Birthday Paradox



http://en.wikipedia.org/wiki/Birthday_paradox

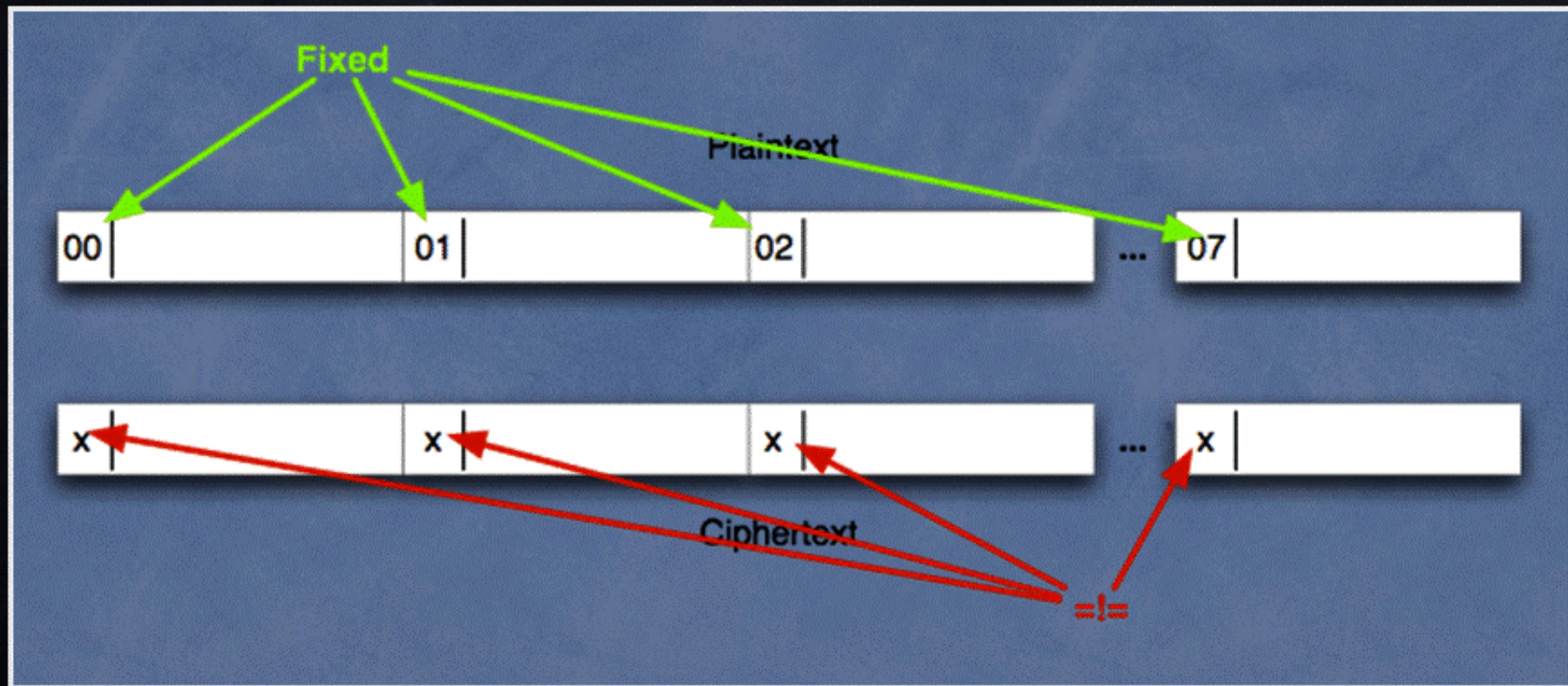
Volume Encryption Challenge

- ④ 512 bytes in, 512 bytes out, all used
 - ④ implicit IV based on sector number
 - ④ constant for each sector
 - ④ No room for additional integrity
 - ④ No ability to determine if there were changes
- ④ If we assume attackers can -
 - - modify ciphertext
 - - rearrange ciphertext
 - - read some (but not all) sectors

CBC Watermark

- If you store my file on your encrypted disk
 - I can detect it
- Naked-IV
 - $IV = \text{sector number}$
- Sectors are allocated in 4k units
 - 8 contiguous sectors starting at $s \equiv 0 \pmod{8}$
 - $IV = s = (U \mid \{0..7\})$
 - Trick is to set $P1 = 0..7$ for each sector
 - Input to encryption will always be $(U \mid 0)$
 - Ciphertext for C1 will be identical for each sector

CBC Watermark

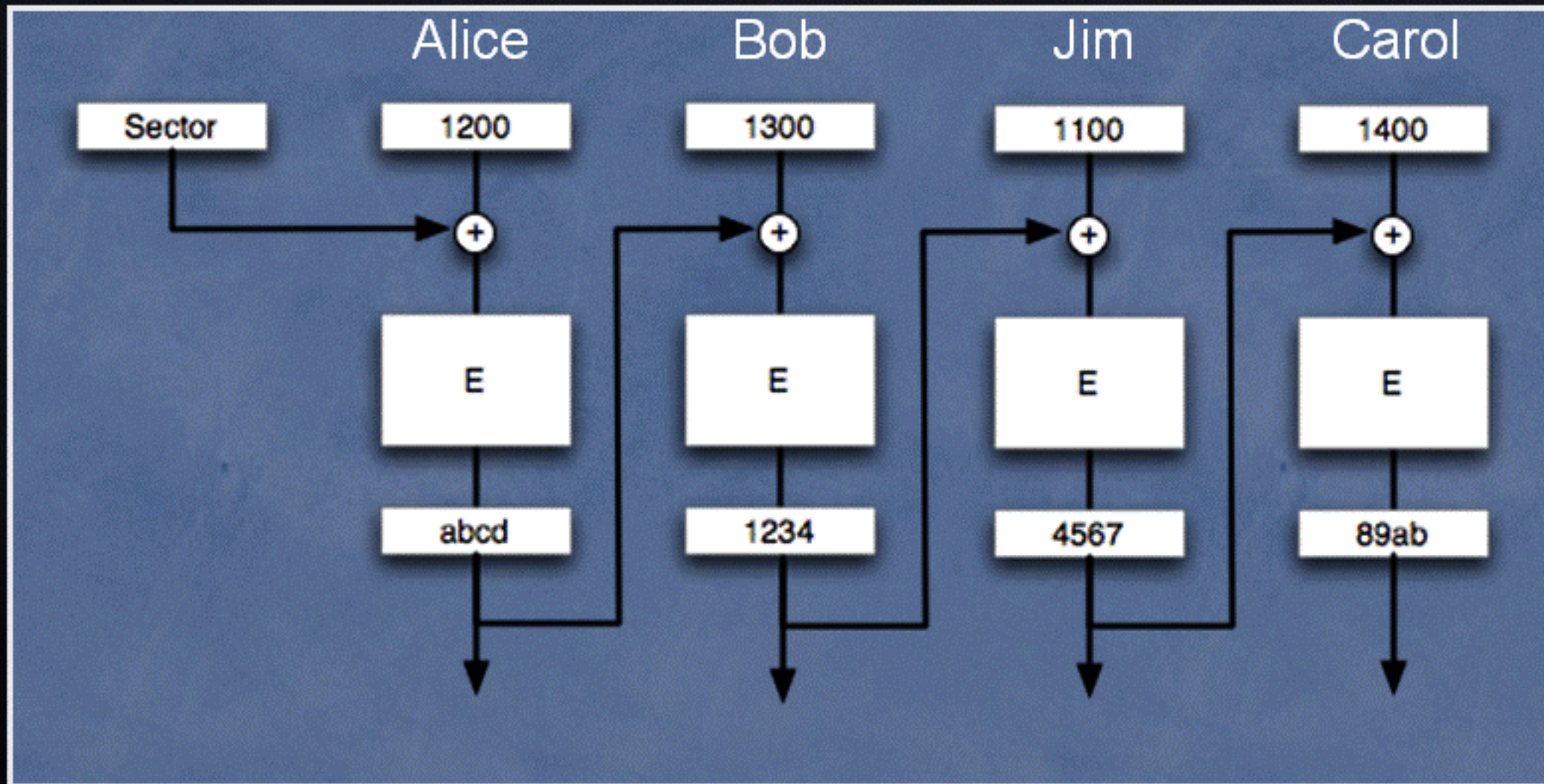


Maleability

- Can the attacker make changes that they understand?
 - Increase my salary?

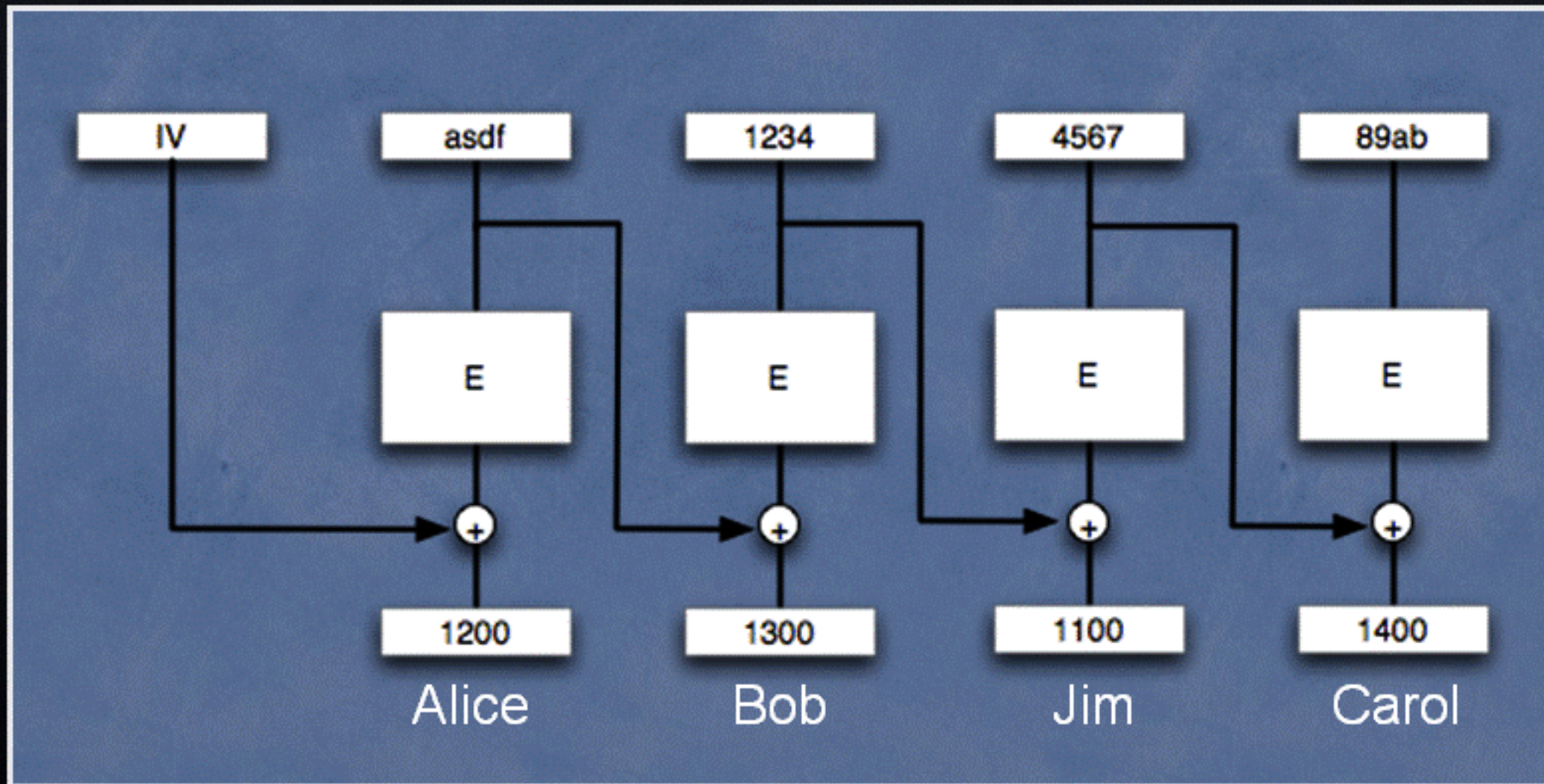
Salary Database

- Encryption



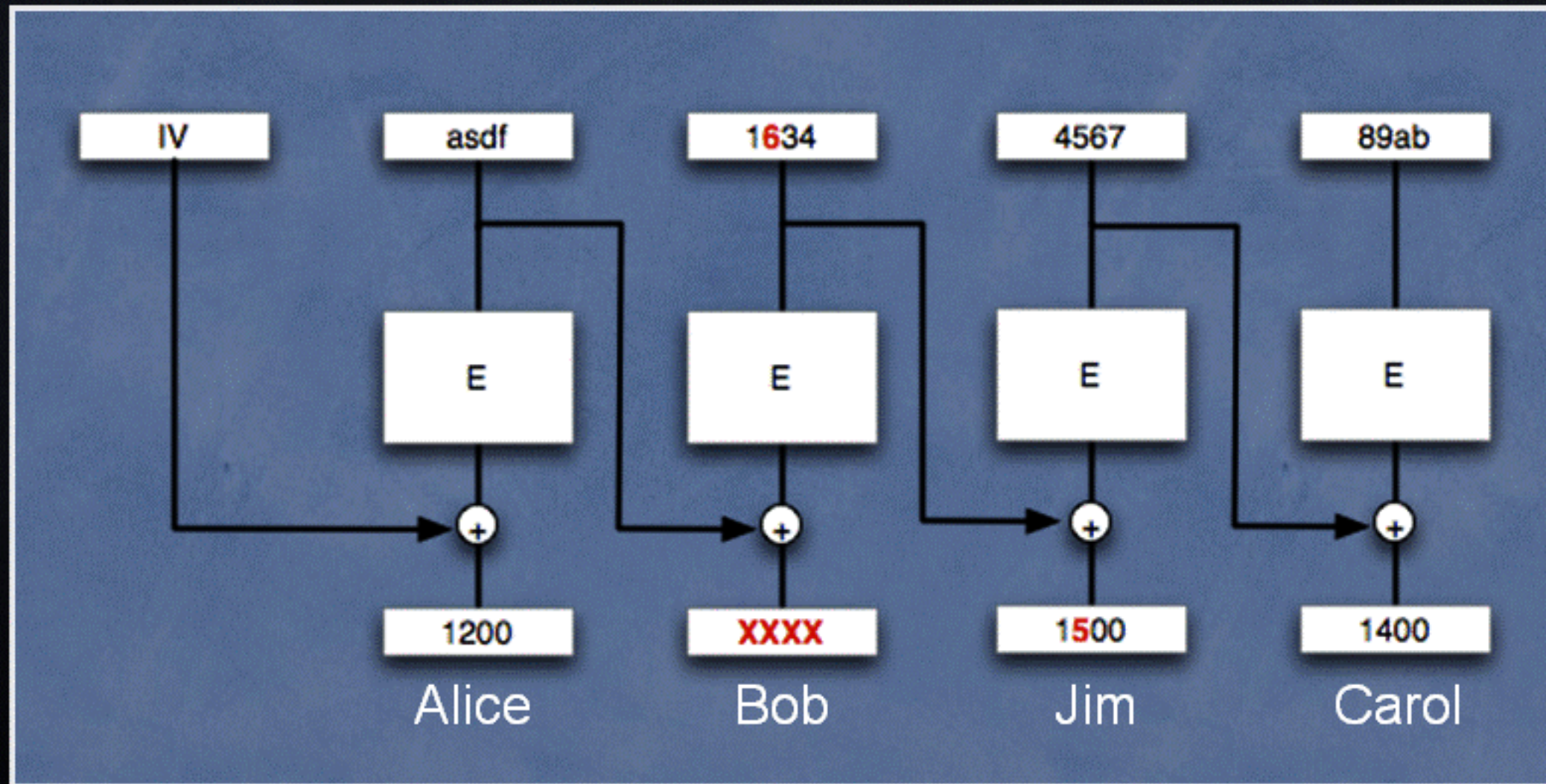
CBC Malleability

- Decryption



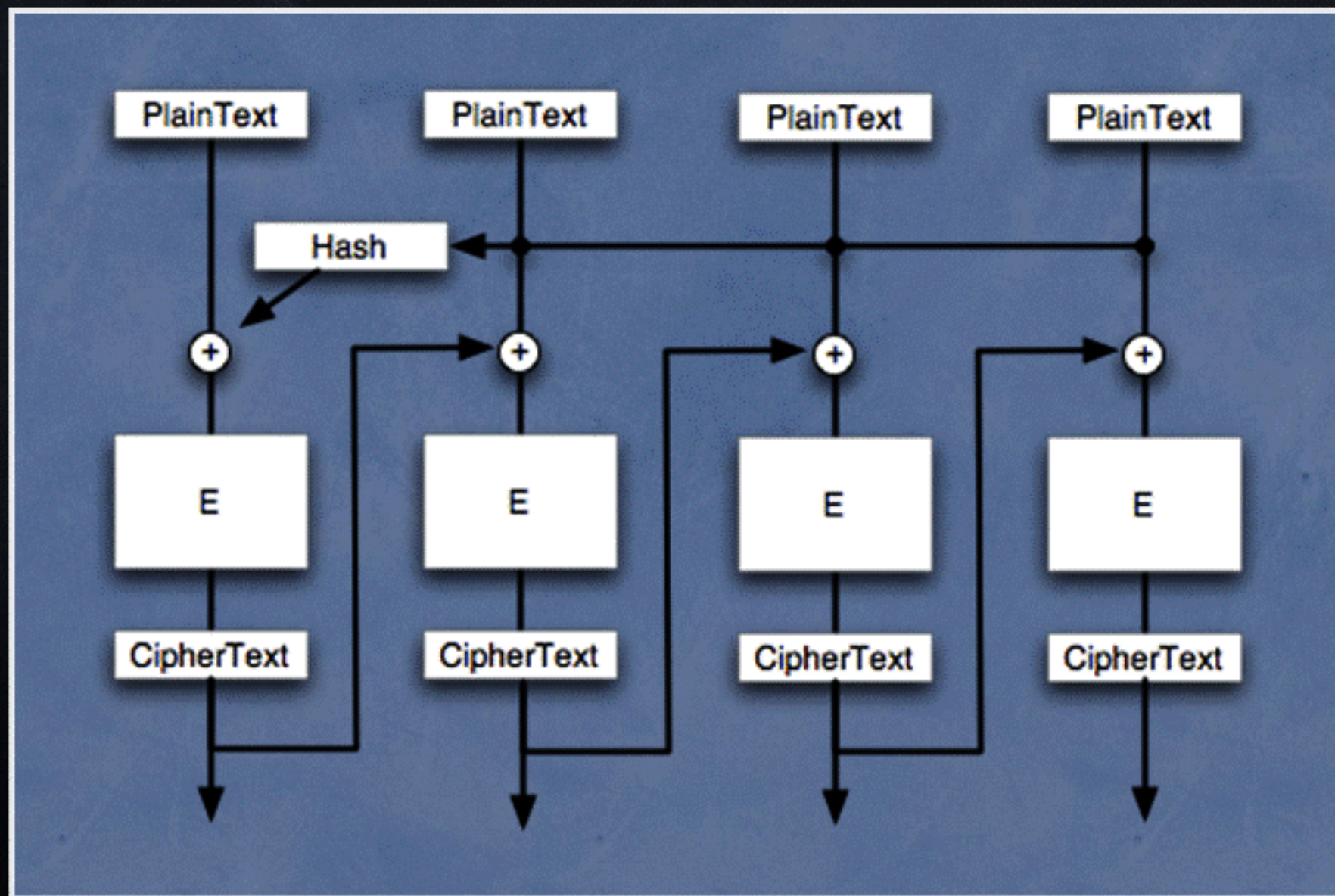
CBC Malleability

- Decryption



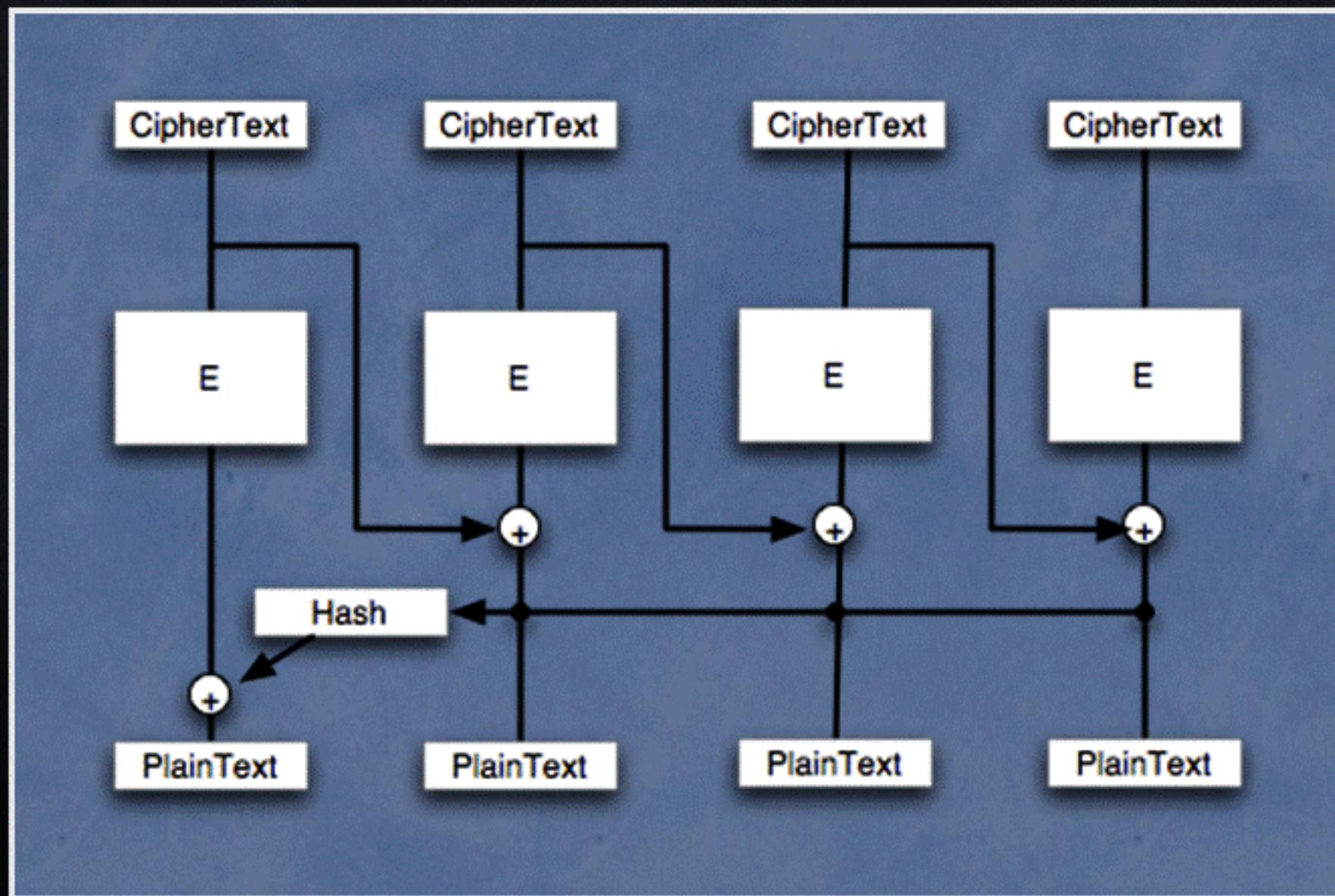
Plumb-IV Encryption

- Uses the data to create the IV

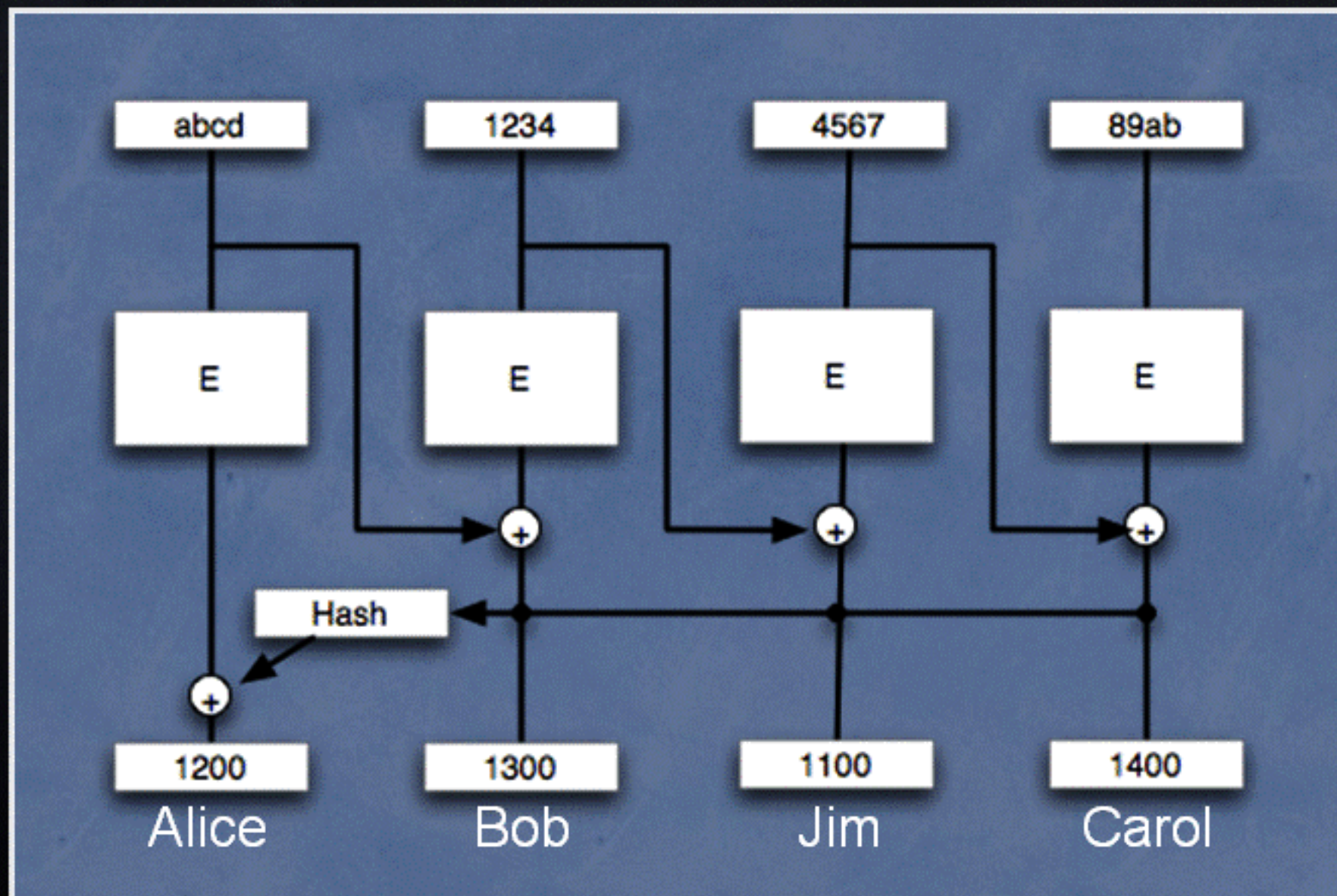


Plumb-IV Decryption

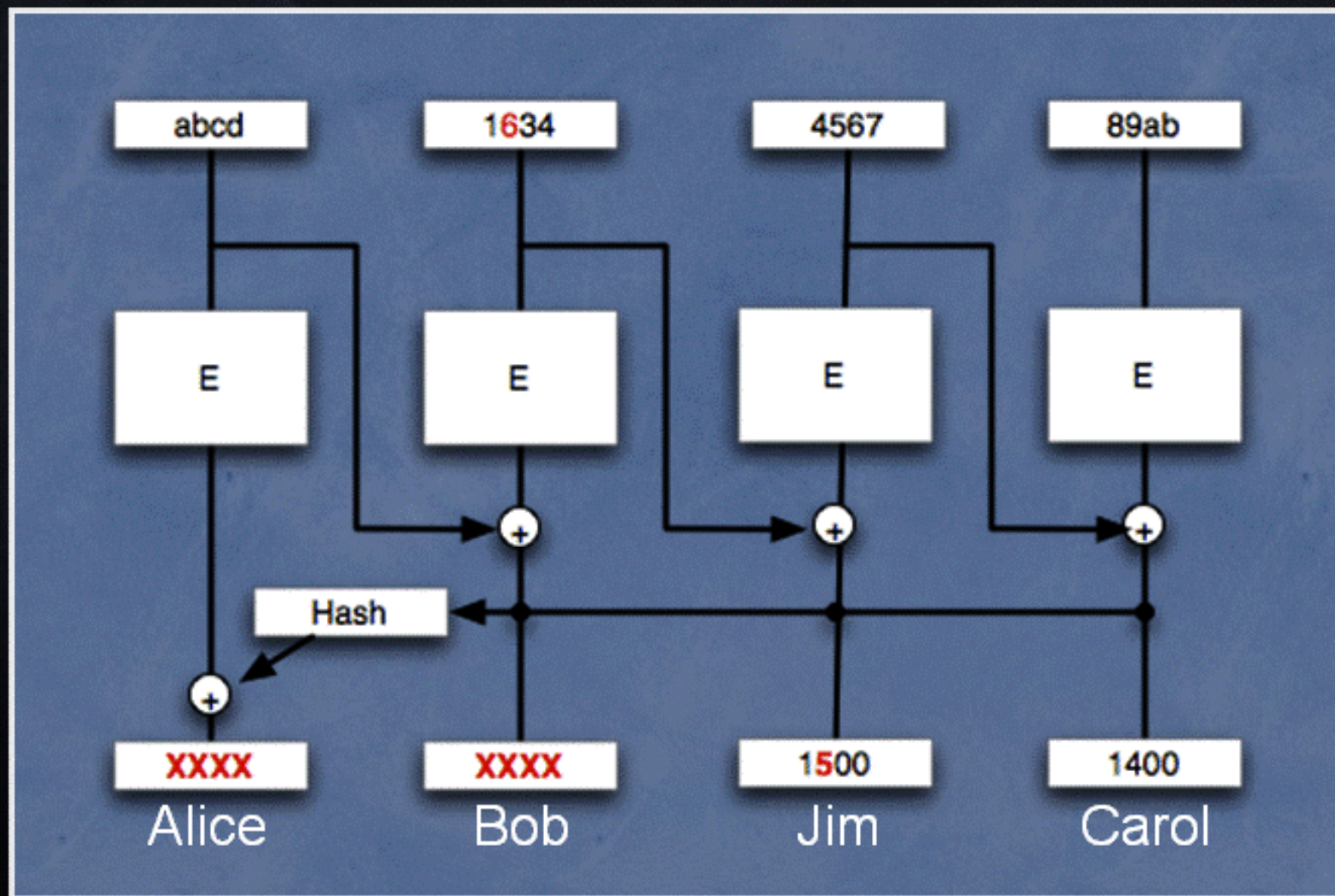
- Uses the data to fix P_1



Plumb-IV Malleability



Plumb-IV Malleability



Sector Move Attack

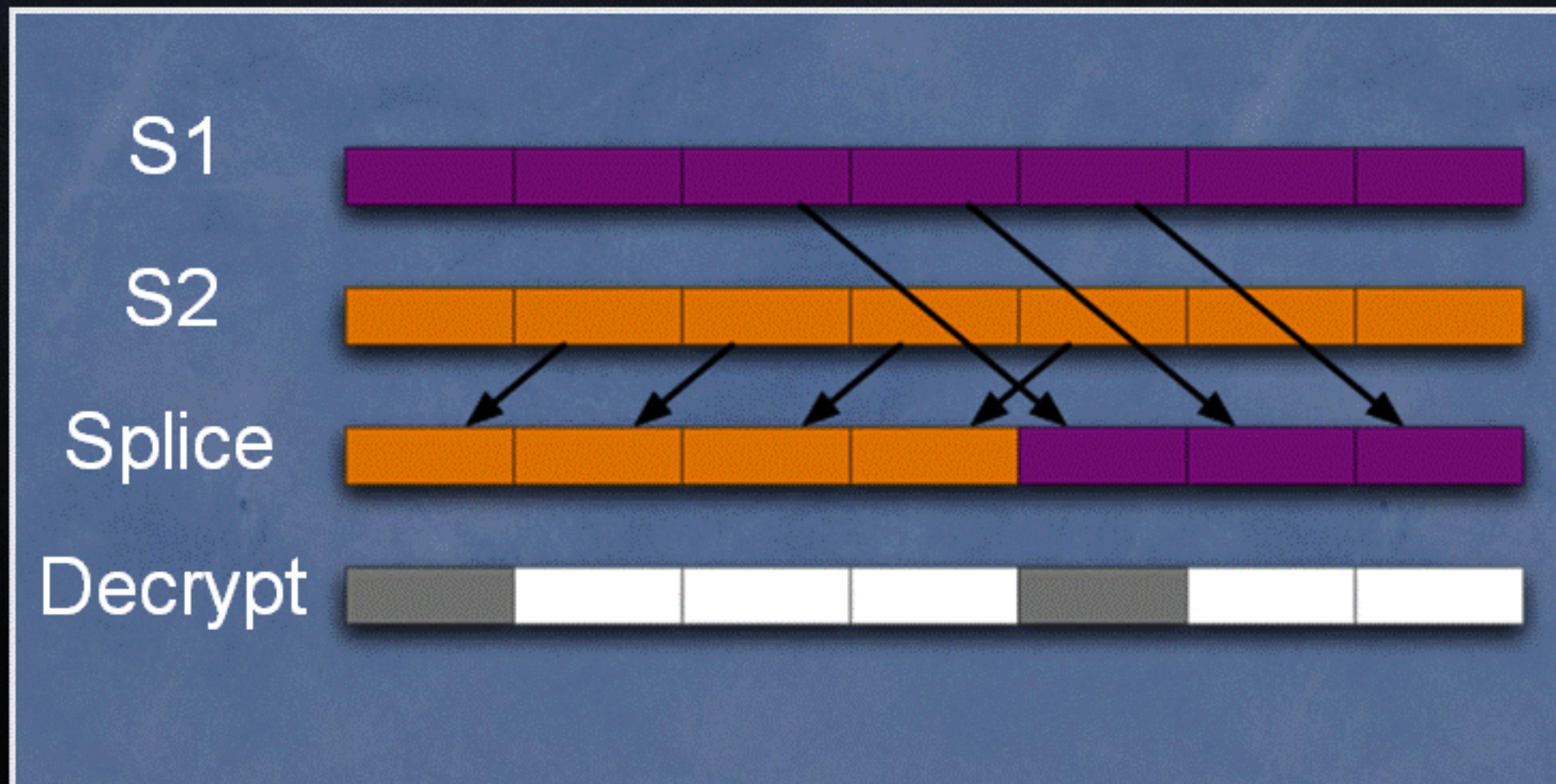
- Assume as guest I can read one sector
 - “my sector”
- Move all other ciphertext to my sector
 - Read all the data on the disk

Movable

- Take one sector and move it to another location
- CBC is “Self Synchronous”
 - Good feature for Communications
 - 2 block error extension
- Move a block $C_{1b\dots nb} \rightarrow C_{1a\dots na}$
 - $P_{2\dots n} = P_{2b\dots nb}$
 - $P_1 = P_{1b} \oplus IV_a \oplus IV_b$
 - If we know the IVs, we can correct (Plain IV)
 - Other modes, lose P_1 (Enc IV, Plumb IV)

Cut and Paste

- Take any 2 sectors and splice them together
 - one block error at splice point
 - before and after correct



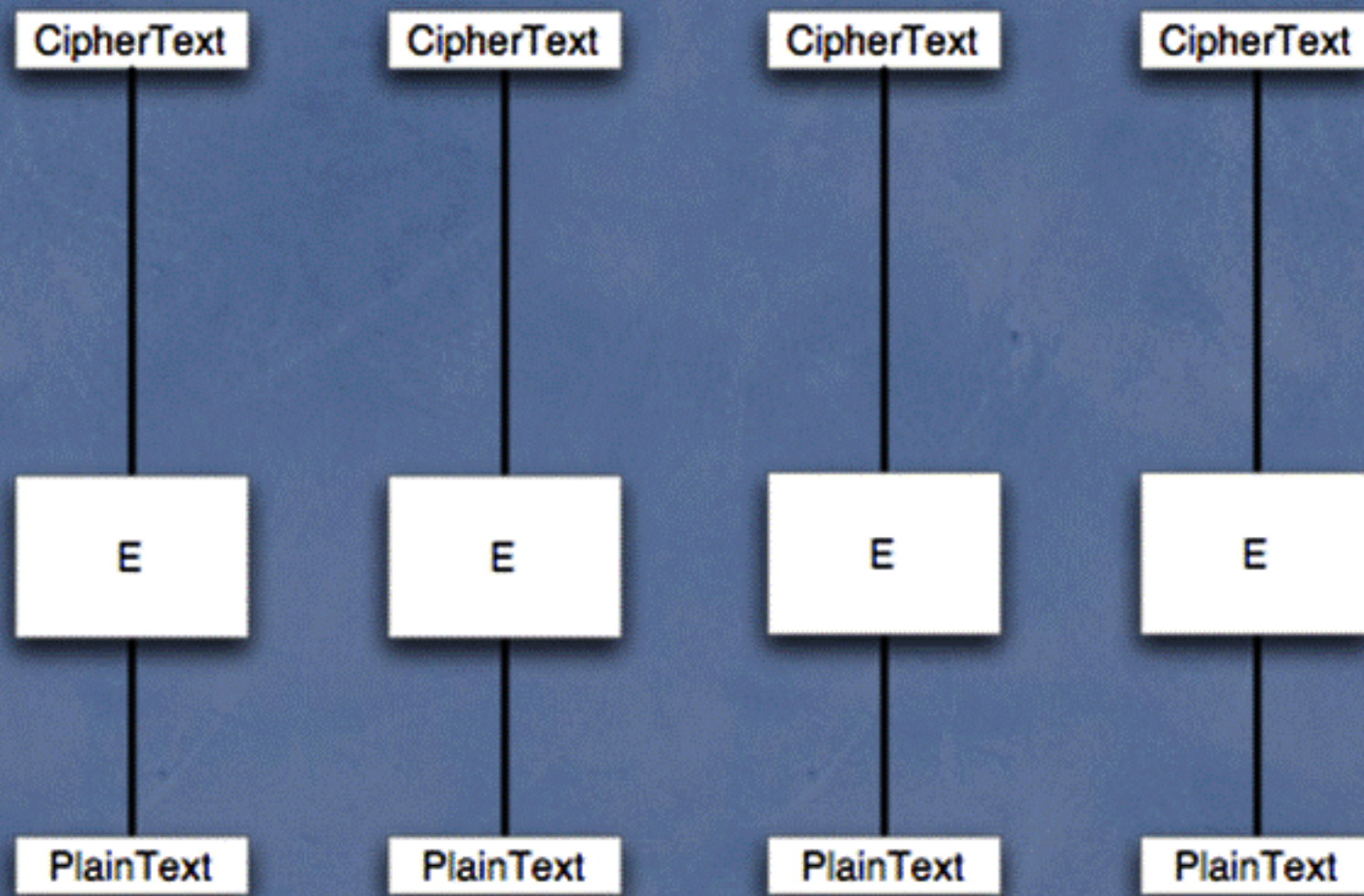
Tweakable Cipher

- Tweakable Block Ciphers
 - Liskov, Rivest, and Wagner
- A new cryptographic primitive, the “tweakable block cipher.” Such a cipher has not only the usual inputs - message and cryptographic key - but also a third input, the “tweak.” The tweak serves much the same purpose that an initialization vector does for CBC mode or that a nonce does for OCB mode. Our proposal thus brings this feature down to the primitive block-cipher level, instead of incorporating it only at the higher modes-of-operation levels. We suggest that (1) tweakable block ciphers are easy to design, (2) the extra cost of making a block cipher “tweakable” is small, and (3) it is easier to design and prove modes of operation based on tweakable block ciphers.

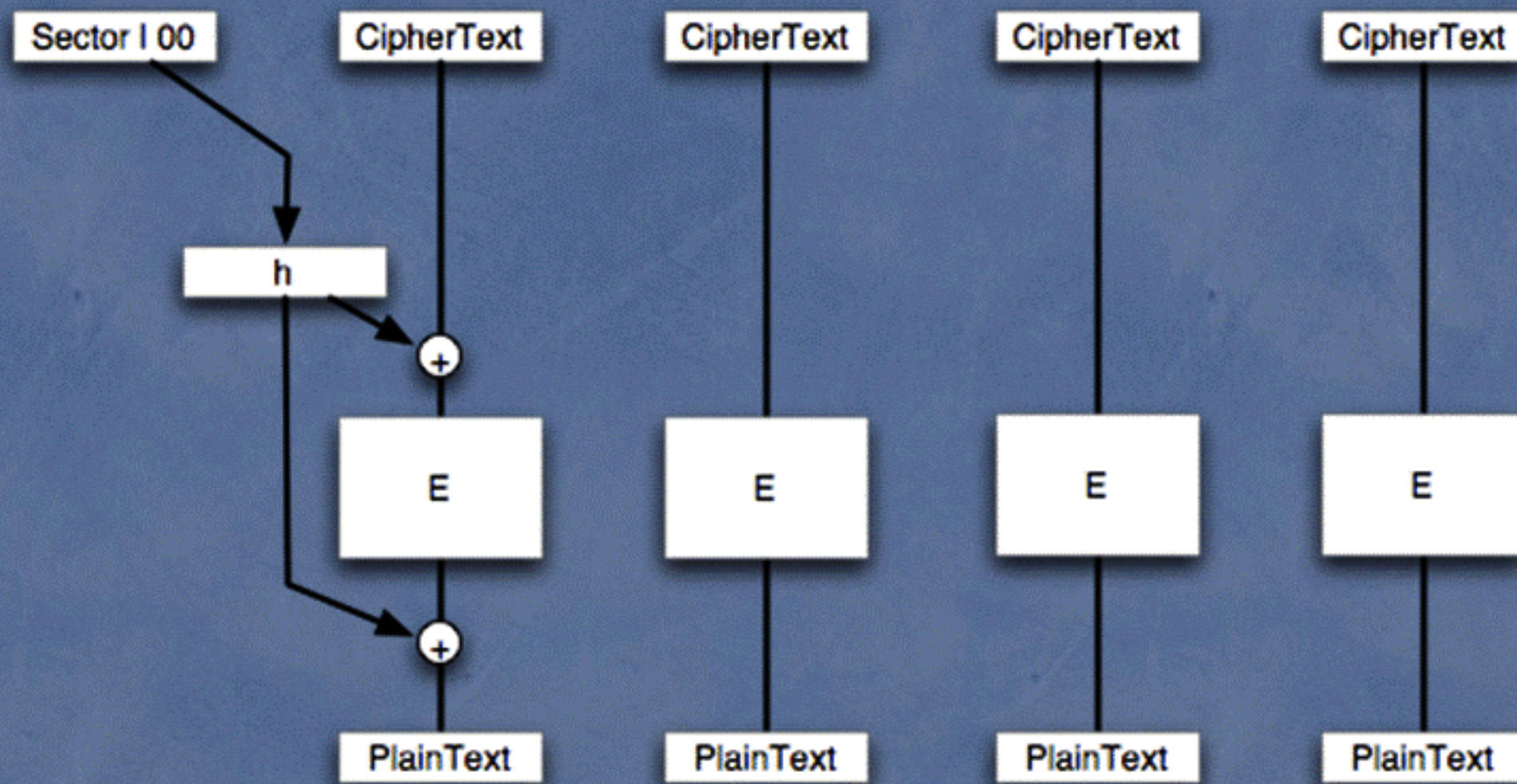
Tweakable Cipher

- Like having a new key for each “tweak” value
 - The permutation a cipher produces with a given key is different for each tweak value
 - Tweak does not need to be secret
- Eliminates dictionary attack
 - when different tweaks are used
- P1619 LRW mode (Kent)
 - Uses keyed hash
 - Tweaked ECB mode

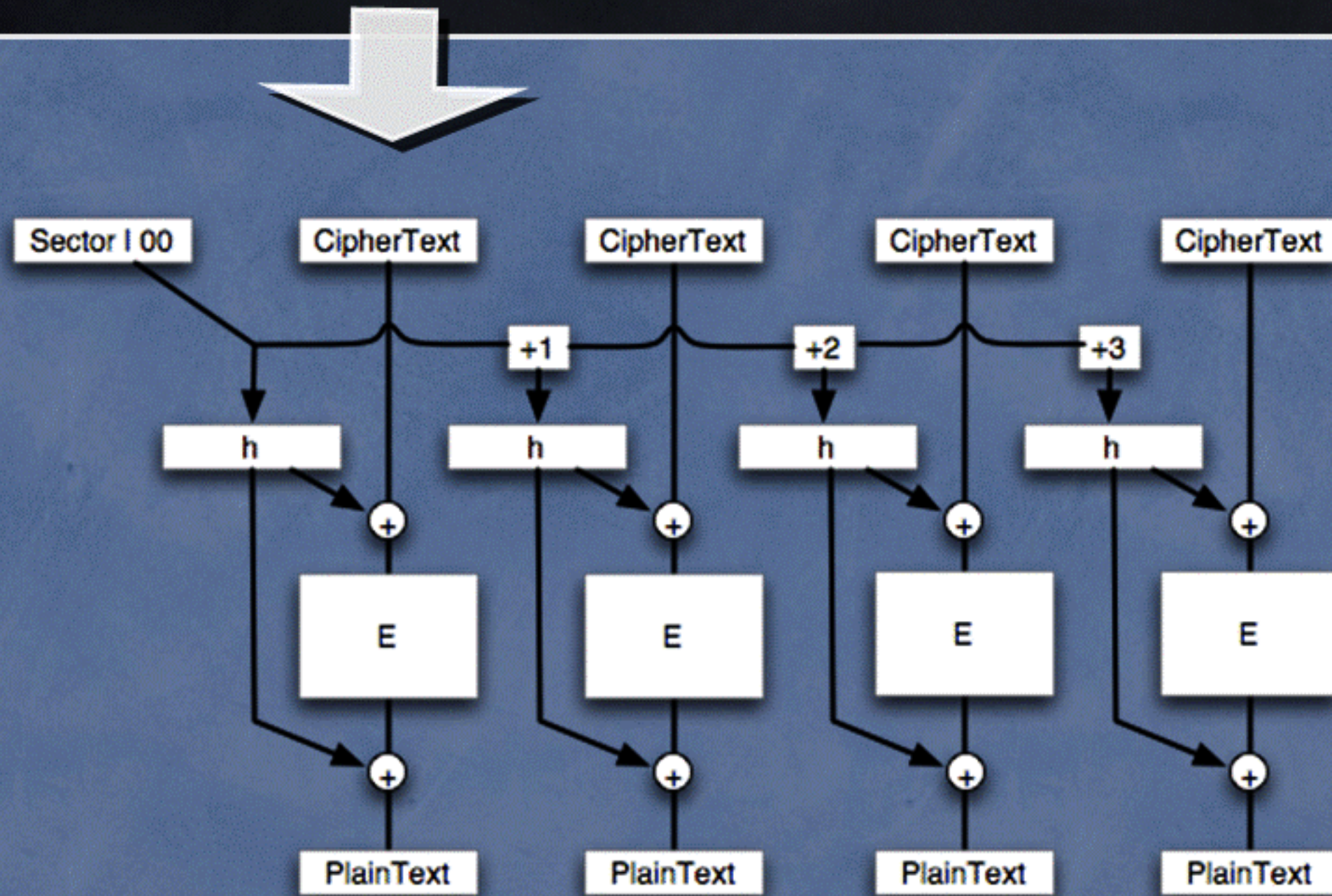
LRW



LRW



LRW



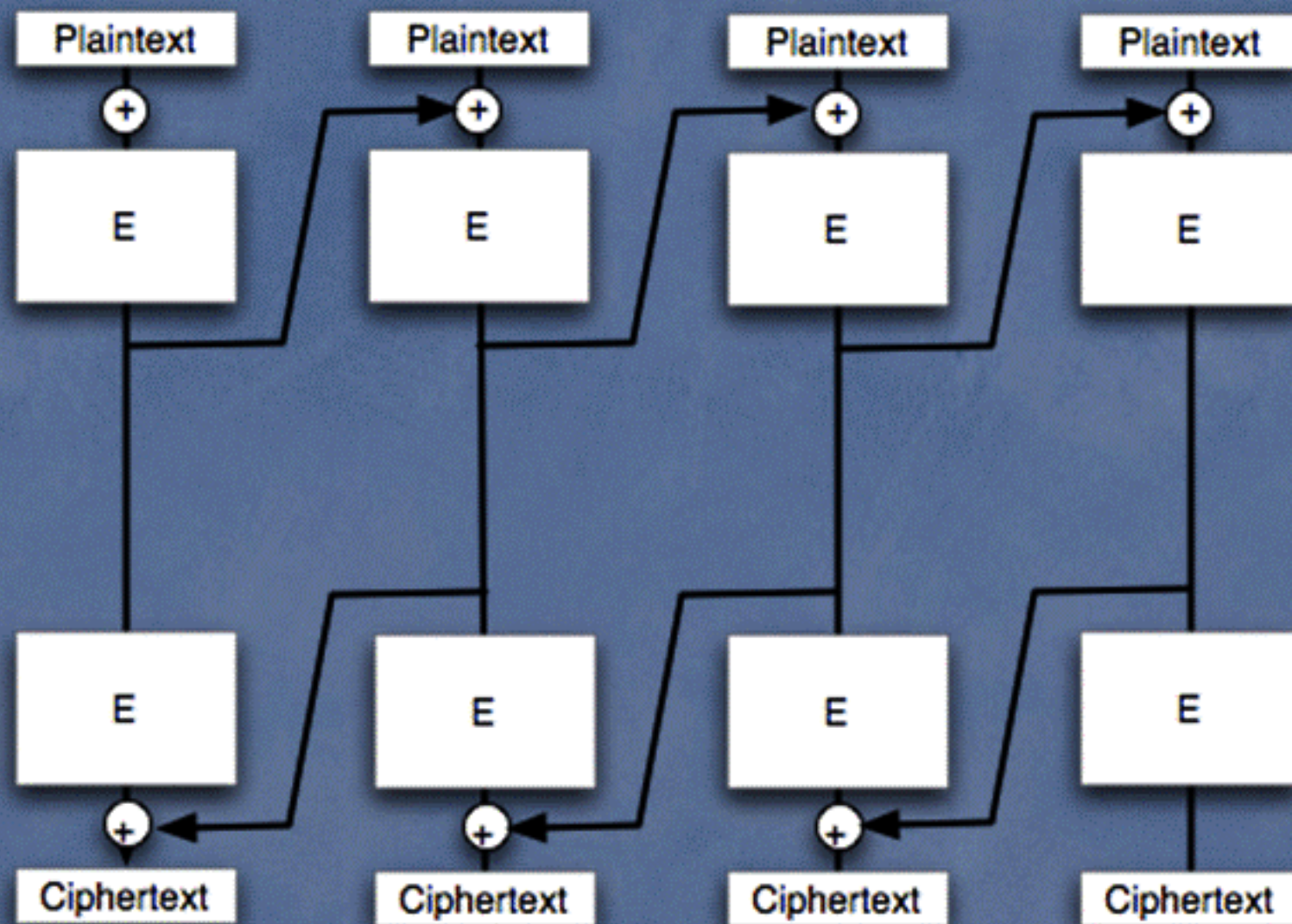
Replay-Block Attack

- Personnel action in database
 - Change back
- LRW, If a previous value is returned
 - Plaintext value returns
- 16 byte chunk (AES)

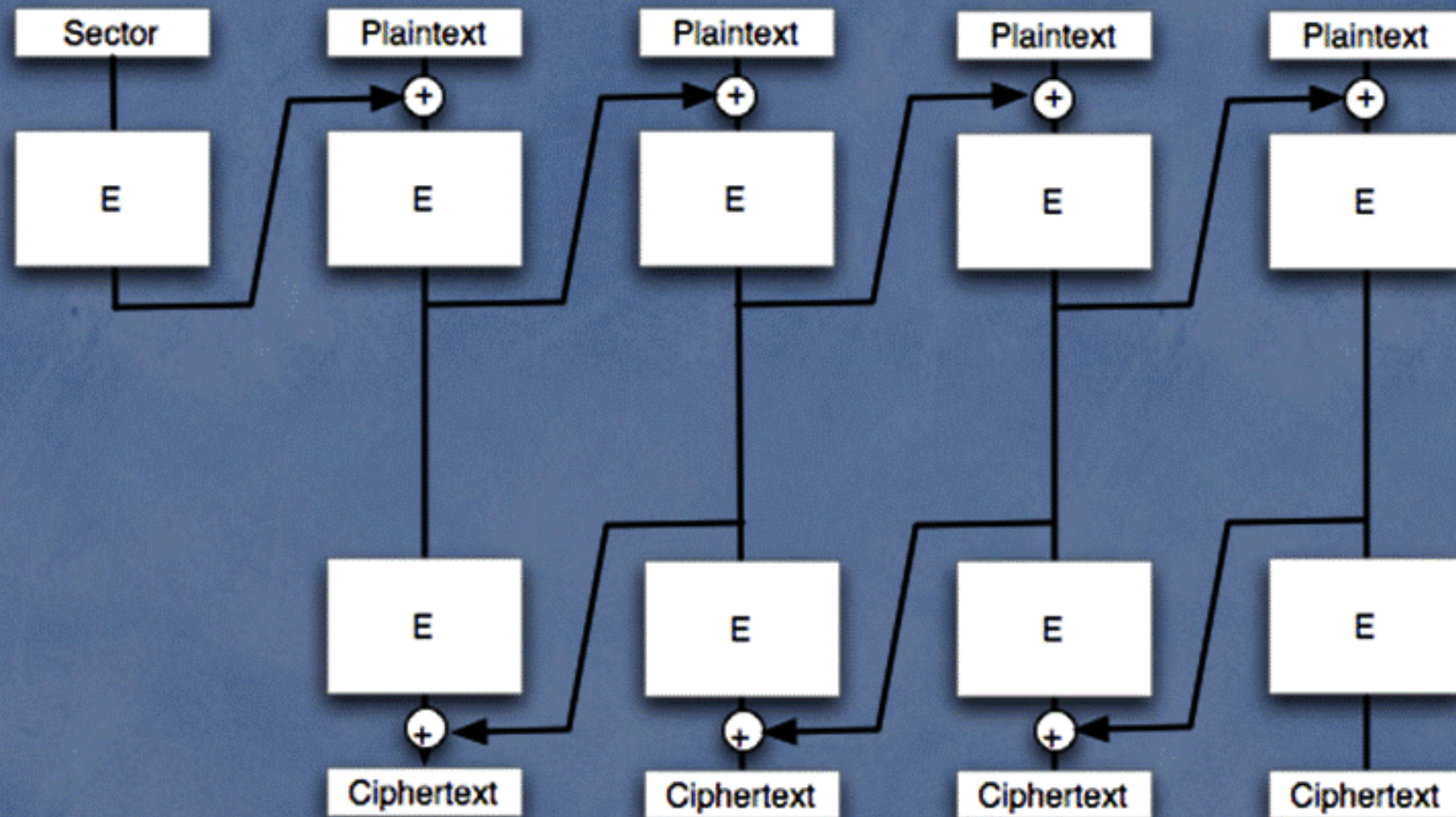
CMC / EME

- A tweakable wide block cipher
 - 512 byte block
 - Permutation of 2^{4096} values
- Made out of another narrow block cipher
 - AES
- Any change anywhere, random result
 - Not malleable
 - Not Splicable
 - Not movable (since tweaked)
- Only remaining vulnerability
 - the entire sector can be returned to a previous value

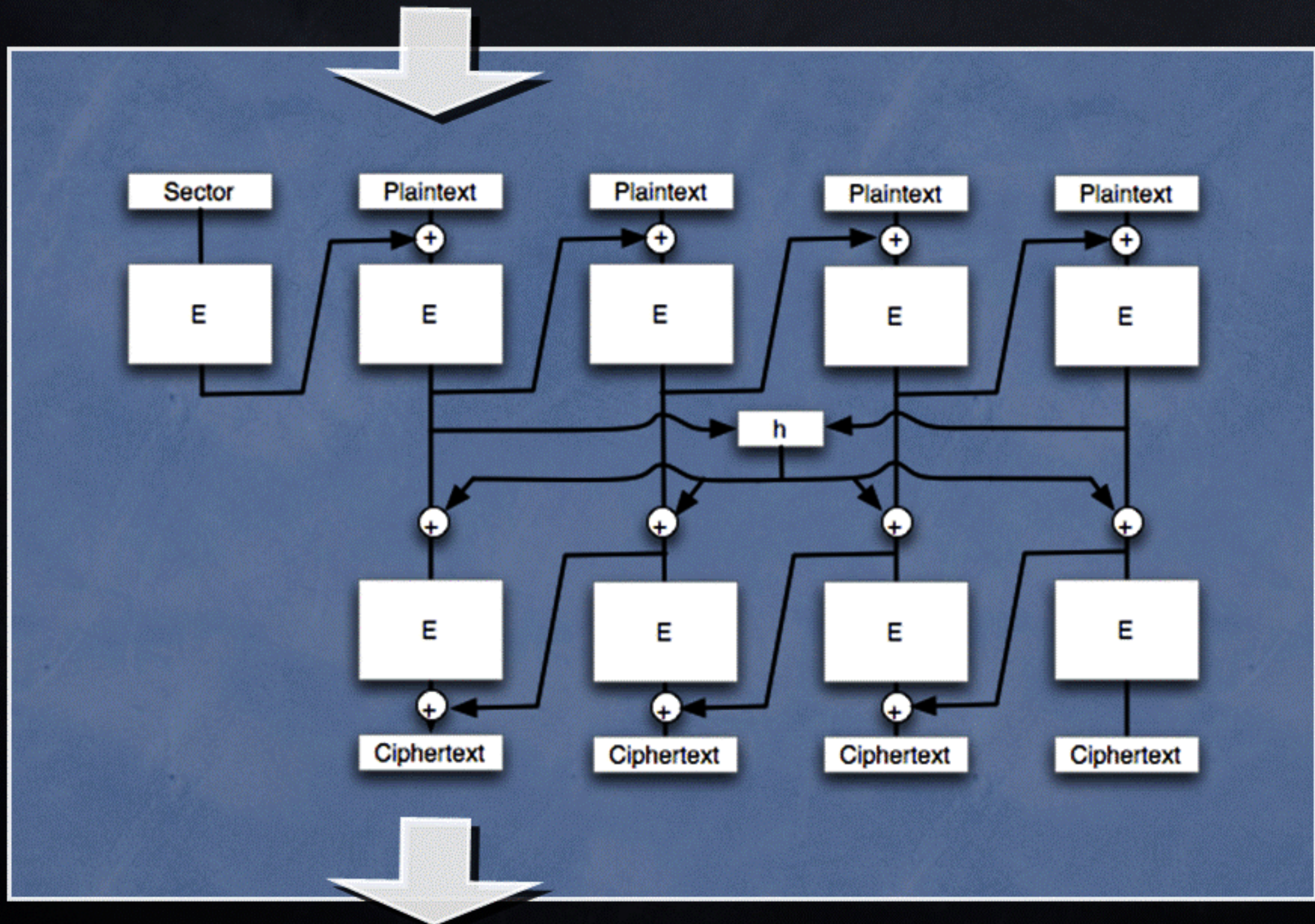
CMC Mode



CMC Mode



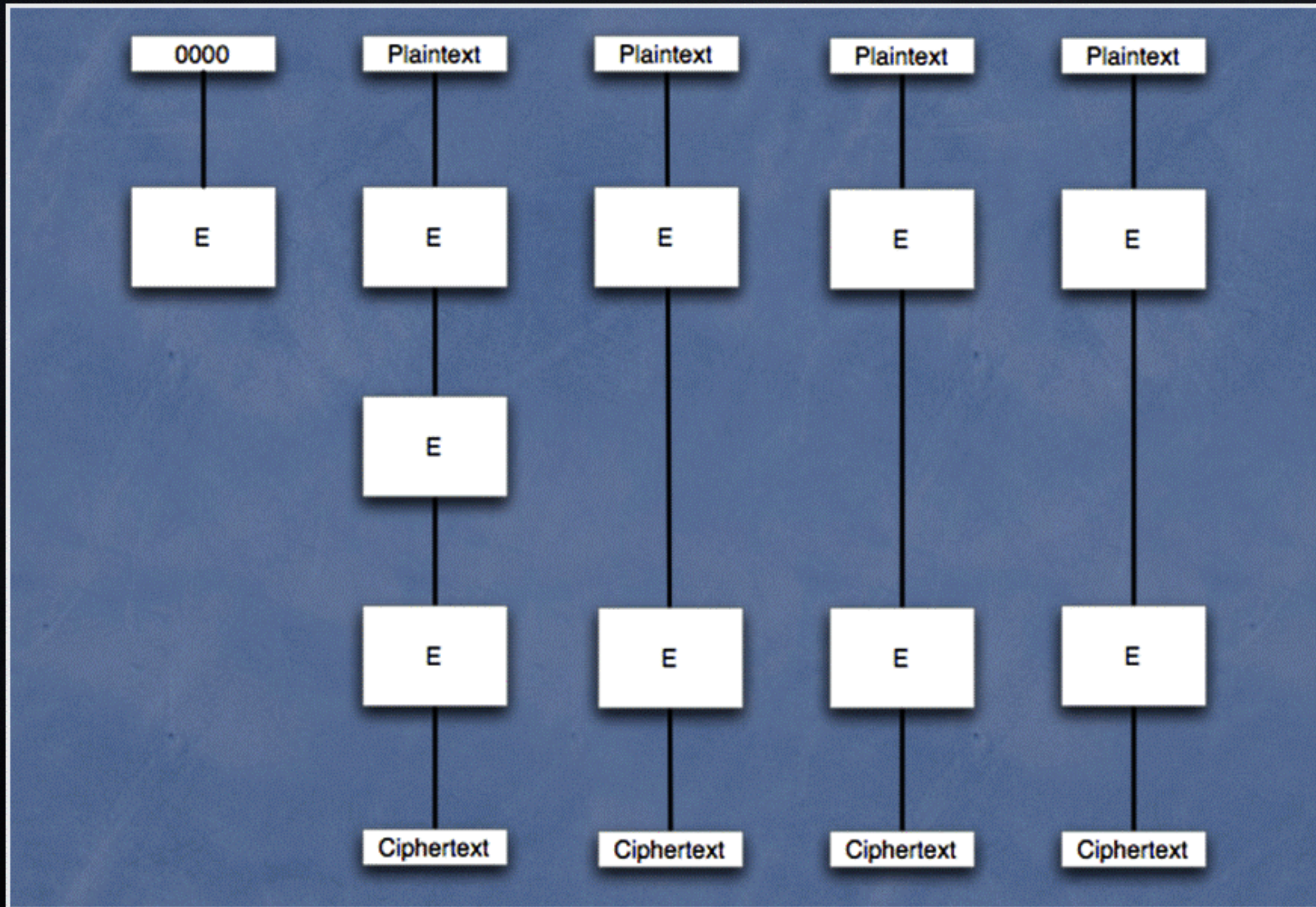
CMC Mode



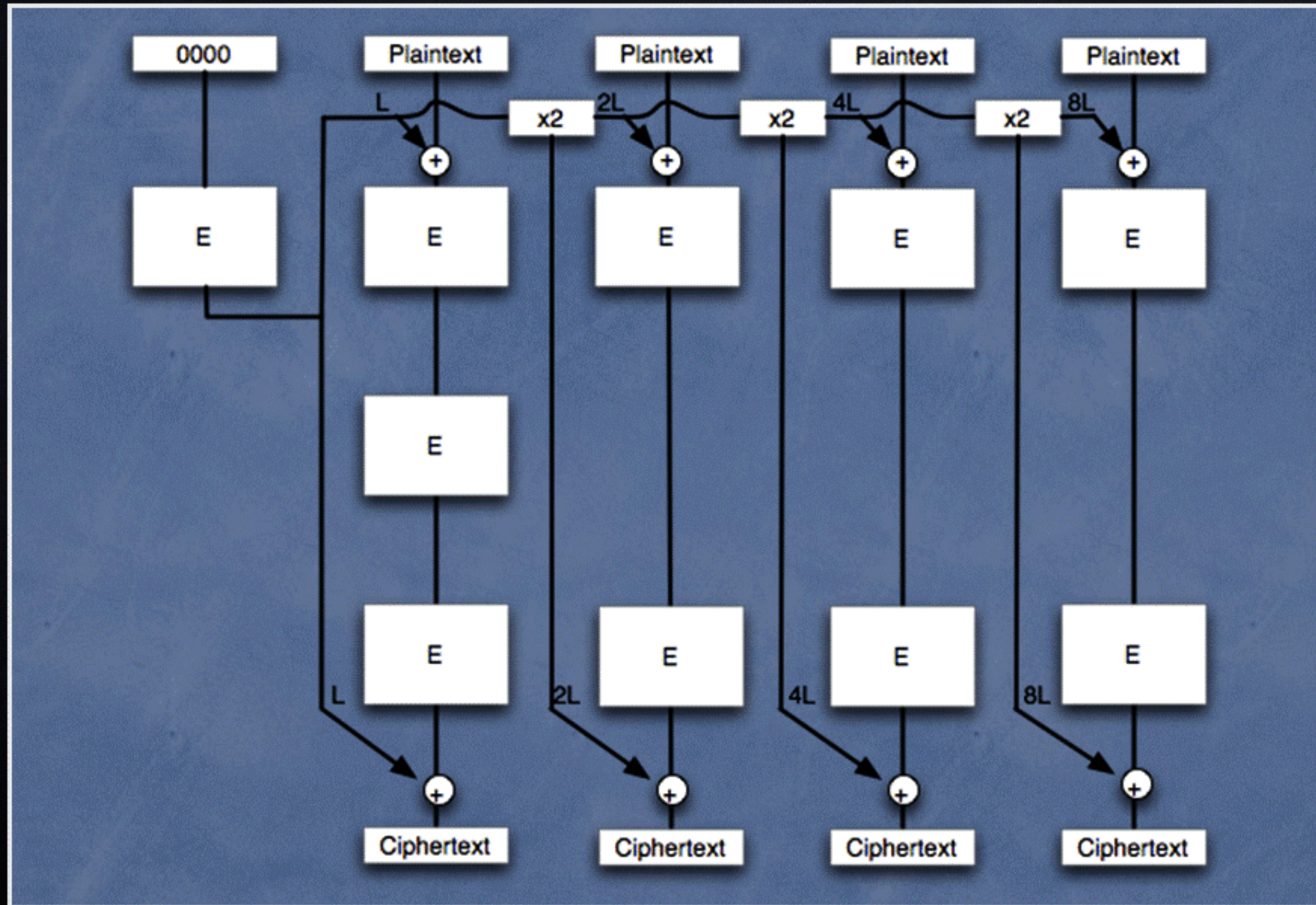
EME Paralleizable Encryption

- CMC meets the requirements
 - But is serial and requires 64 encryptions
 - Min length 33 encryptions
- EME meets the requirements
 - Requires 65 encryptions
 - Min length 3 encryptions

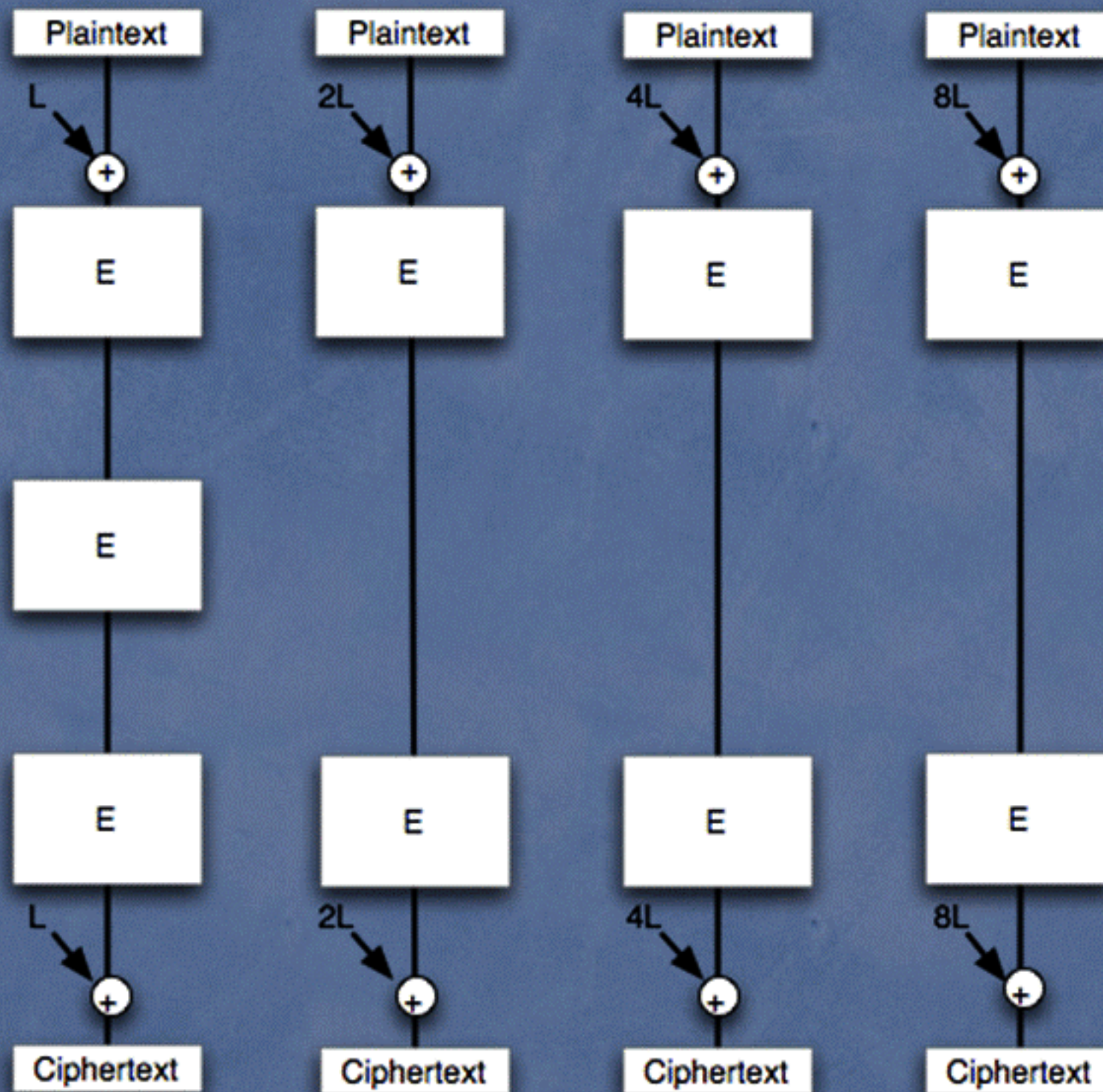
EME



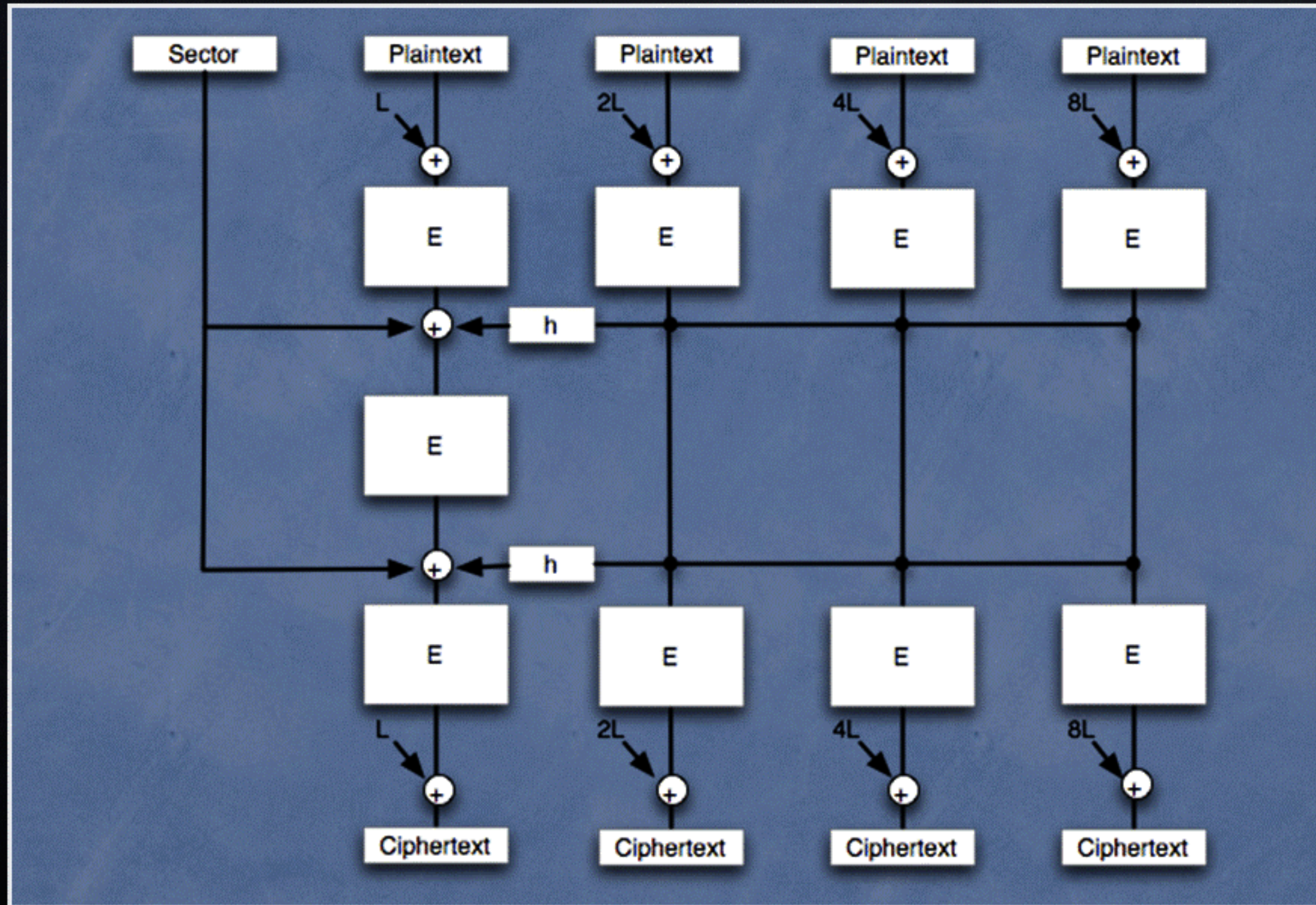
EME



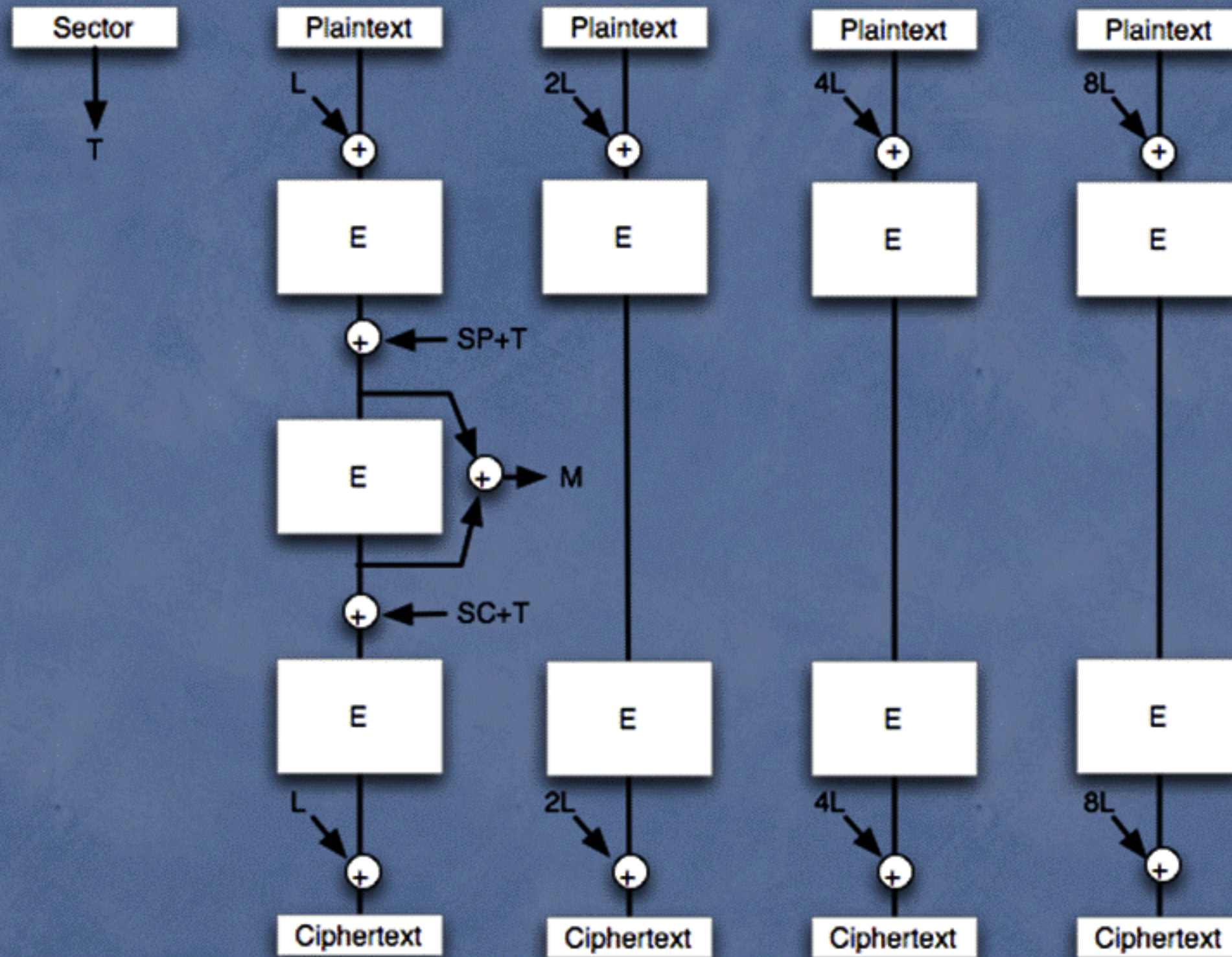
EME



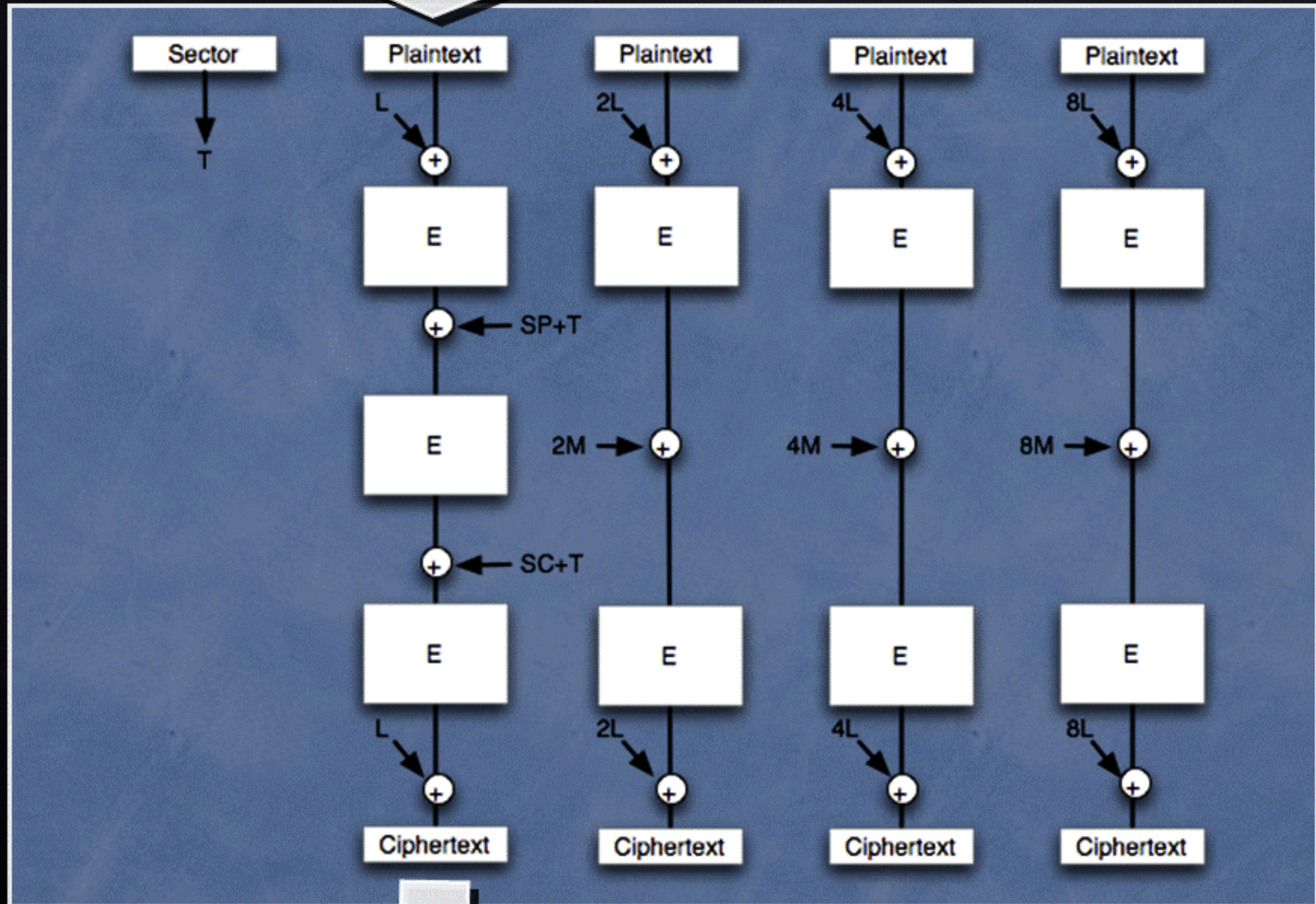
EME



EME



EME



Modes	Leaks	Watermark	Malleable	Movable	Replay-Block	Replay-sector
ECB	Yes	Yes	No	Yes	Yes	Yes
Plain IV	Yes	Yes	Yes	Yes	Yes	Yes
Enc IV	Yes	No	Yes	Yes	Yes	Yes
Plumb-IV1	Yes	?	Yes	Yes	Yes	Yes
LRW	No	No	No	No	Yes	Yes
CMC	No	No	No	No	No	Yes
EME	No	No	No	No	No	Yes

Destruction of Data in a hurry

- ① ~~Deleting the file~~
- ① ~~Over writing the data~~
- ① ~~Shoot the drive~~
- ① ~~Security Erase~~
- ① ~~De-Gaussing~~
- ① ~~Melting~~
- ① Encryption?
 - ① Delete the key, delete the data

Tape Encryption

- Tape allows extra data
 - Addition of Nonce and check values

Conclusion

- Standards
 - Standards provide many eyes to look at a problem
 - Mistakes are subtle
 - Modes are more complicated than they seem
- The future
 - Disk volume encryption will be common
 - Built into the OS
 - Boot from protected drives
 - Modes other than CBC will be used