Symmetric Encryption

Applied Cryptography

The Four Primitives

- There are four primitives which are considered the building blocks of digital cryptography
    - Random Number Generation
    - **Symmetric Encryption**
    - Asymmetric Encryption
    - Hash Functions
- These primitives get combined to achieve the CIA (confidentiality, integrity, authentication) goals of cryptography

Symmetric Encryption

"We've devised a new security encryption code. Each digit is printed upside down."

https://www.glasbergen.com/ngg_tag/encrypted/nggallery/image/business-cartoons-data-security-security-information-encryption-encrypt-information-data-encryption-encrypt-data-information-security-cartoon-safety-internet-security-keeping-safeguard-saf-1

Symmetric Encryption

- Sometimes referred to as **SKC** - Secret Key Cryptography, symmetric encryption is primarily used to achieve confidentiality
- There are many algorithms for symmetric encryption, built to operate as **stream ciphers** or **block ciphers**
- The key used to encrypt and decrypt is sometimes called a "**shared secret**" or "**secret key**" because both parties to the process must have the key and the message confidentiality is reliant on the key remaining secret
- When 2 parties create and share keys for future communications, they are "**pre-shared secrets**" and they get re-used
- Symmetric encryption is fast and secure, and is used whenever there is a viable method of sharing a key

Secret Key or Shared Secret

- A sequence of bytes, often printable ascii, used to both encrypt and decrypt a message

Pre-shared Secret

- A secret key which has been distributed to both parties prior to communications starting

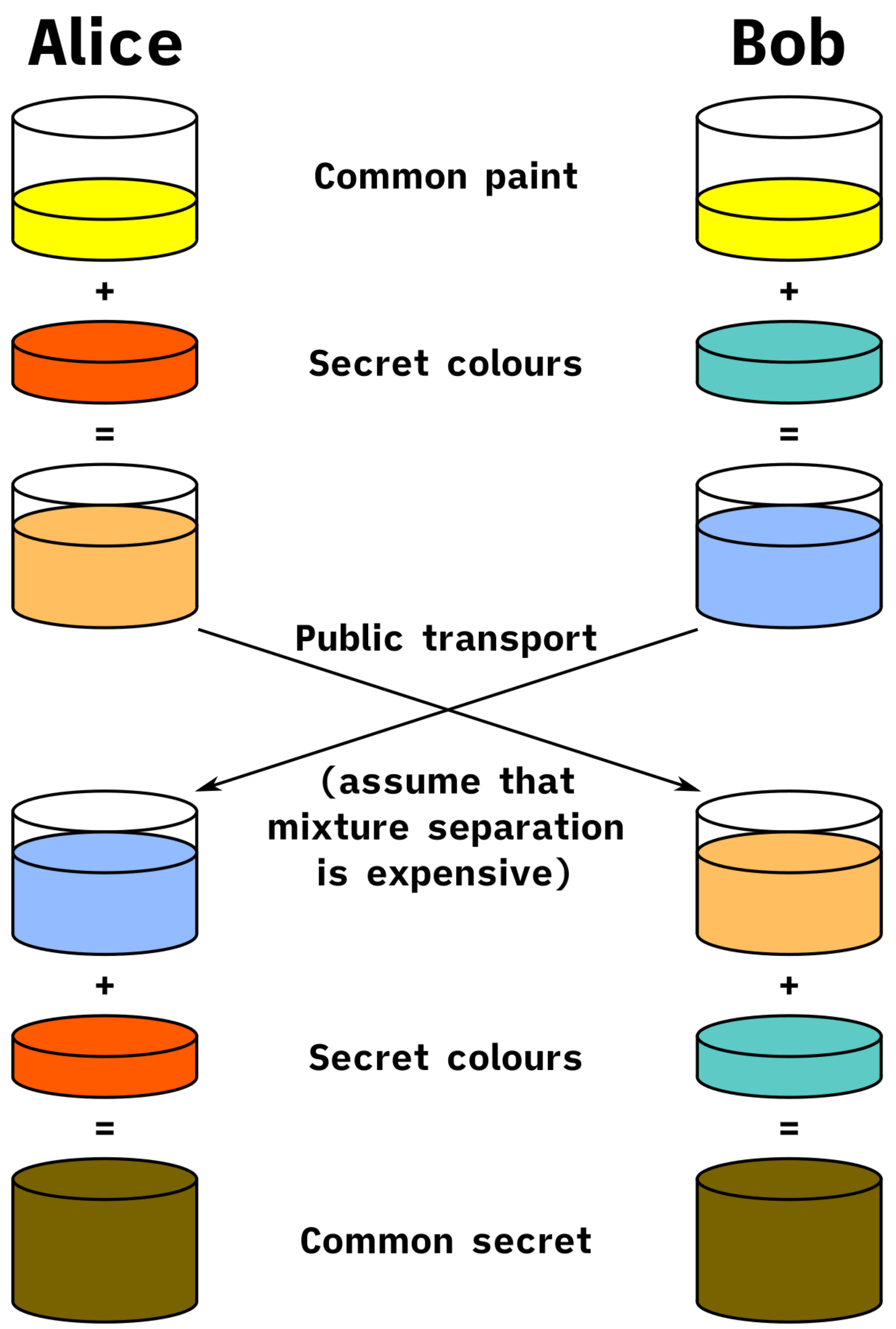
Key Exchange

https://www.kingston.com/en/usb-flash-drives/datatraveler-2000-encrypted-usb-flash-drive

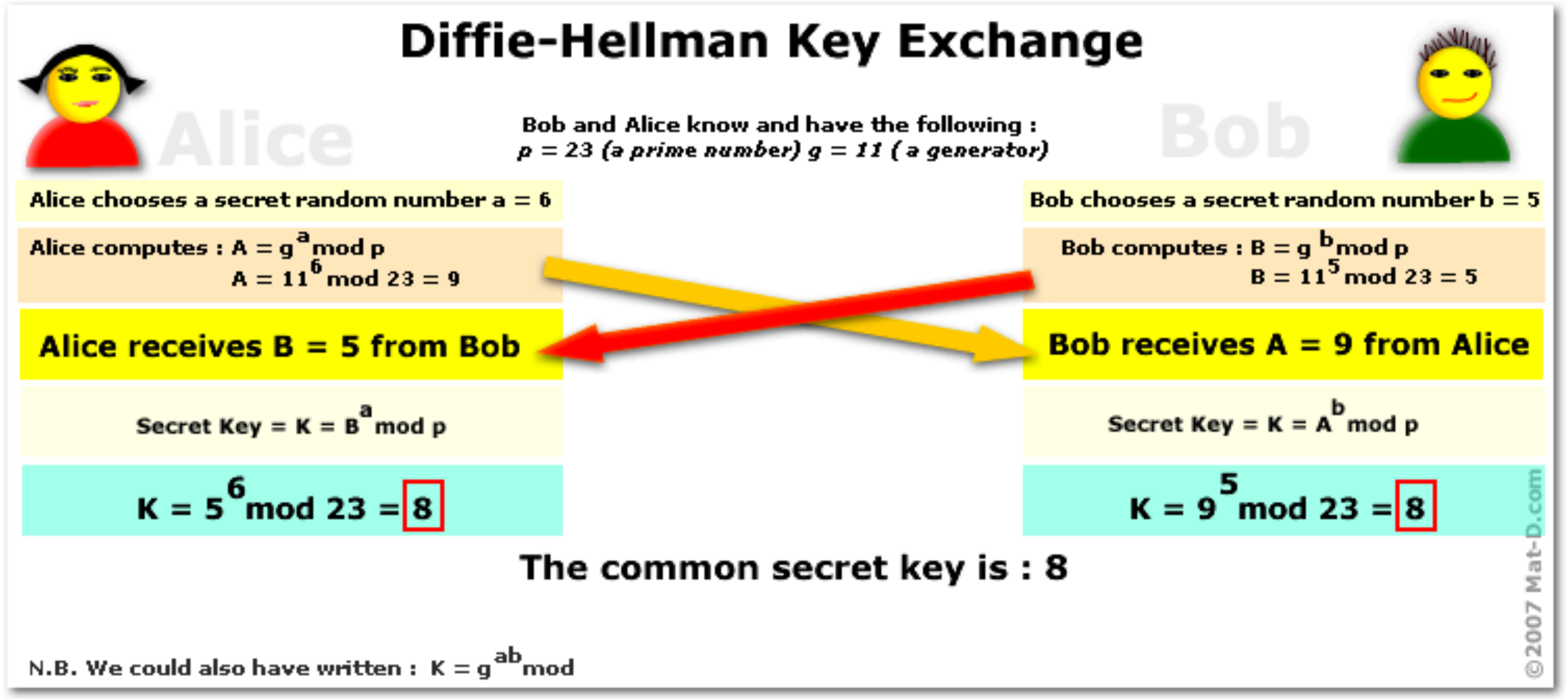
- Before encryption can be done, keys must be created and provided to both the encrypting and decrypting parties
- Keys must be kept secret because ciphers are published
- The method used to digitally exchange keys can be manual between parties or automated using a key exchange protocol
- Key distribution, control, and management is the reason that we don't just do all encryption with OTP

Key Exchange
Diffie-Hellman Protocol

- The Diffie-Hellman protocol is the main protocol that has been used for a long time to automatically and securely exchange single-use (session) keys at time of use between co-operating parties
- It takes advantage of the challenges of working with very large prime numbers to make it computationally impractical to extract the keys from a captured exchange in a useful timeframe



https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange


Diffie-Hellman Key Exchange

Alice

Bob

Bob and Alice know and have the following :
$p = 23$ (a prime number) $g = 11$ (a generator)

Alice chooses a secret random number $a = 6$

Alice computes : $A = g^{a} \bmod p$
$A = 11^{6} \bmod 23 = 9$

Alice receives $B = 5$ from Bob

Secret Key = $K = B^{a} \bmod p$

$K = 5^{6} \bmod 23 = \boxed{8}$

Bob chooses a secret random number $b = 5$

Bob computes : $B = g^{b} \bmod p$
$B = 11^{5} \bmod 23 = 5$

Bob receives $A = 9$ from Alice

Secret Key = $K = A^{b} \bmod p$

$K = 9^{5} \bmod 23 = \boxed{8}$

The common secret key is : 8

N.B. We could also have written : $K = g^{ab} \bmod$

©2007 Mat-D.com

http://internetokracy.appspot.com/crypto1

Stream Ciphers

- Stream ciphers encrypt/decrypt arbitrary lengths of data very fast and are easy/cheap to implement in hardware
- The simple ciphers we have reviewed so far are stream ciphers
    - they encrypt one plaintext character at a time
    - the keystream is simply the key repeated as necessary
- Better quality (longer and harder to predict) keystreams are desirable for useful encryption

Keystream

- A piece of data that is repeated as necessary to provide the unpredictable bits required by encryption algorithms when encrypting plaintext and decrypting ciphertext

KeyStreams

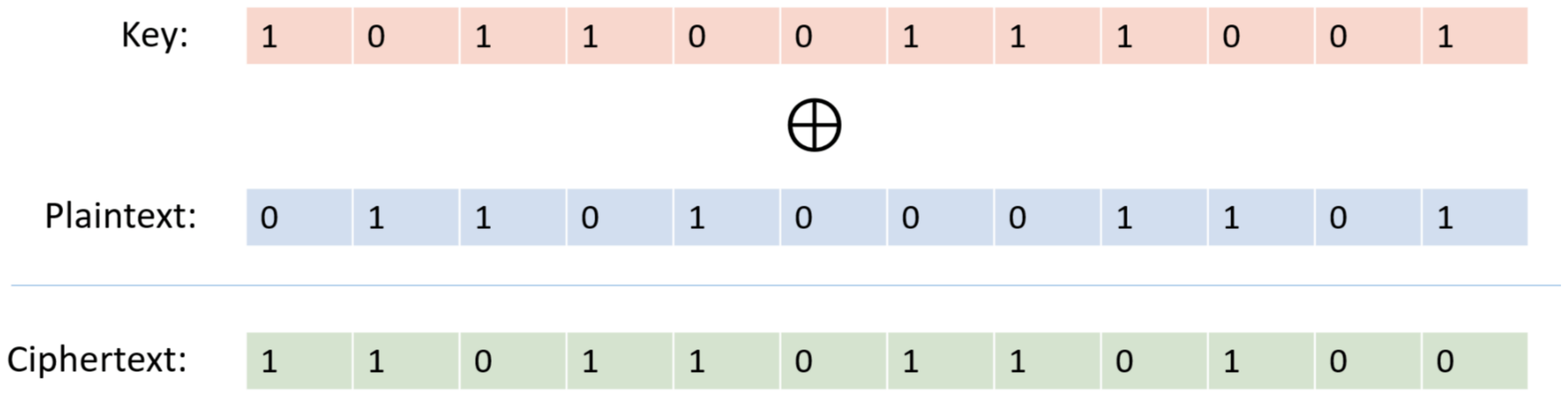
- To get an unpredictable sequence of bits for the keystream, a short key that is human-friendly is used as a seed for a PRNG resulting in a longer sequence of difficult to predict bits
- PRNGs are mathematical functions, so if the other party to the encryption uses the same PRNG with the same seed (key), they will get the same sequence of bits to use for the decryption keystream
- The longer keystream is harder to attack than the base key would be
- An extra value generated randomly at the time of encrypting the plaintext is often included with the key when seeding the PRNG
- This is to create more possible ciphertexts for the same plaintext/key combination in order to thwart some cryptanalysis techniques
- This extra value is called an Initialization Vector and does not need to be kept secret

Initialization Vector (IV)

- A randomly generated data block mixed with the key or plaintext prior to encryption in order to stymie some cryptanalysis techniques

The Role of Exclusive-Or

- Digital stream ciphers combine a keystream using XOR with the plaintext to encrypt, and the same series of bits using exclusive-or to decrypt ciphertext - the longer the series, the more difficult it is to search the keyspace to find the key

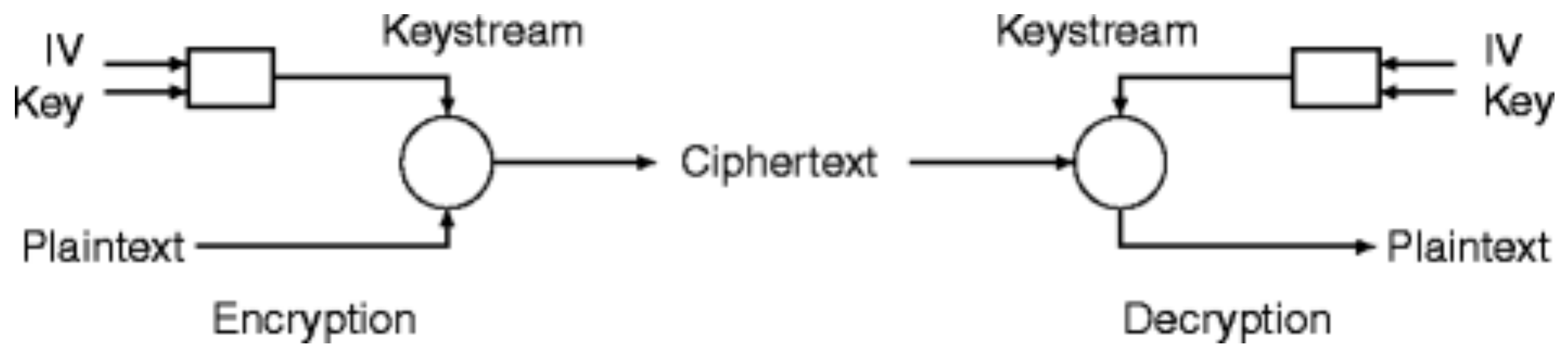


Key:	1	0	1	1	0	0	1	1	1	0	0	1
						$\oplus$						
Plaintext:	0	1	1	0	1	0	0	0	1	1	0	1
Ciphertext:	1	1	0	1	1	0	1	1	0	1	0	0

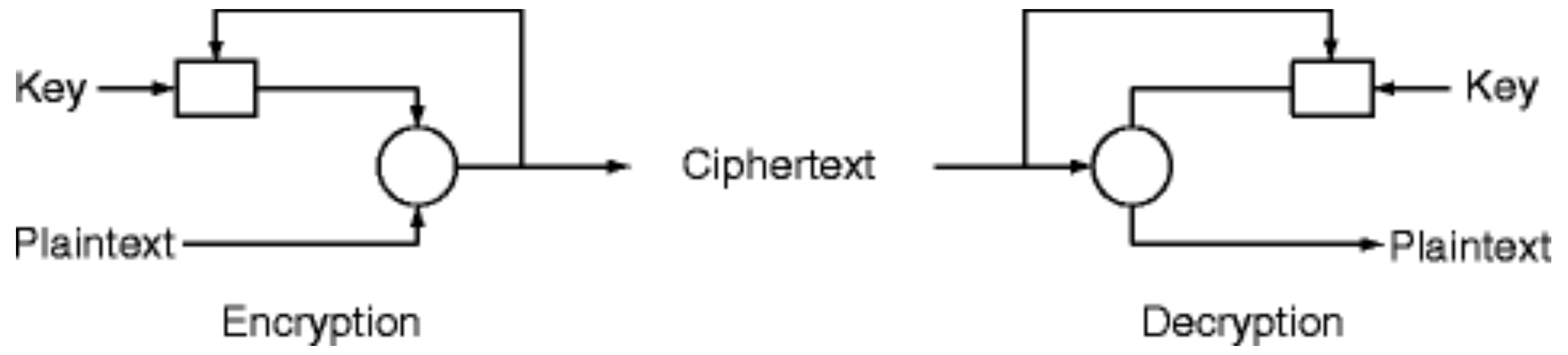
XOR

- Exclusive-or
- A boolean operation on a pair of bits yielding a result bit which is a 1 only if the input bits differ

Synchronous vs. Self-synchronizing Ciphers



https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-5906-5_376



https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-5906-5_376

- A synchronous cipher is one where each piece of generated ciphertext (a bit or byte) in a stream is not dependent on any other piece of ciphertext in the stream
- Synchronous ciphers can be resistant to corruption - one corrupt piece of ciphertext does not affect any other ciphertext
- Self-synchronizing ciphers generate ciphertext with every block of ciphertext dependent on the values of previous ciphertext
- Self-synchronizing ciphers produce ciphertext which is more complex and in theory harder to crack, but require reliable transport
- Synchronous stream ciphers enable encrypting/ decrypting at arbitrary indices into the stream - enables direct access in a stream for resilience, corrupting specific items of a stream enables active attacks

Popular Stream Ciphers

- There are many stream ciphers in use today
- Rivest Cipher 4 (RC4) has been used as the basis for much of the symmetric encryption in use over more than 3 decades, including as a base for wifi encryption
- AES block ciphers operating in streaming modes offer better speed and protection over RC4 and have been very popular
- Salsa20 is a pure streaming algorithm with better speed
- ChaCha, a variation on Salsa20, is rapidly becoming the most popular stream cipher due to its combination of speed, security, and resource consumption

Block Ciphers

- Block ciphers encrypt small blocks of data at a time
- AES (Rijndael) is the most popular block cipher, succeeding DES and 3DES which are no longer computationally secure
- AES uses a 128 bit block size, encrypting 16 bytes at a time and producing 16 bytes of output for each 16 bytes of input
- Block ciphers operate in one of several modes and most use an Initialization Vector, some use nonces
- Because it has streaming modes, AES has been a very popular stream cipher, although ChaCha is overtaking it for popularity

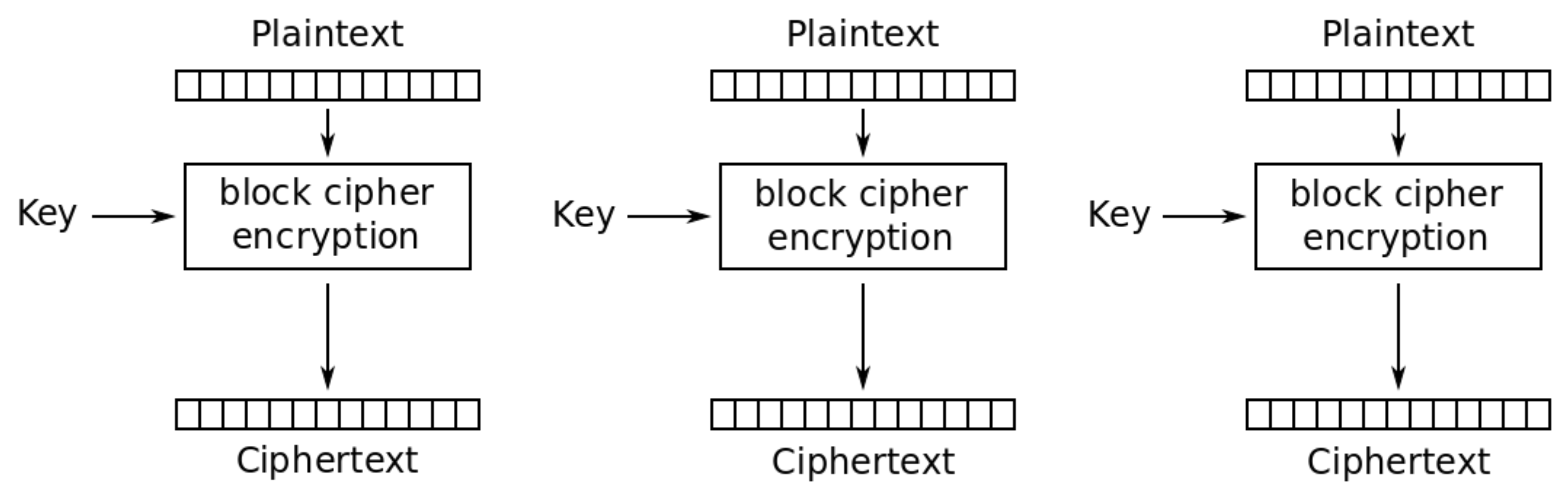
Block Mode

- A technique of using a block cipher to encrypt arbitrarily-long plaintext, effectively making it a stream cipher

Block Cipher Naming

- The name or designation of a deployed block cipher has 3 parts written using their acronyms and dashes to separate the parts:
  - Cipher name
  - Key length in bits
  - Block mode
- E.G.
  - AES-256-CTR is the AES cipher using a 256 bit key and Counter (CTR) block mode
  - AES-192-GCM is the AES cipher using a 192 bit key and Galois/Counter (GCM) block mode

Block Ciphers
ECB mode

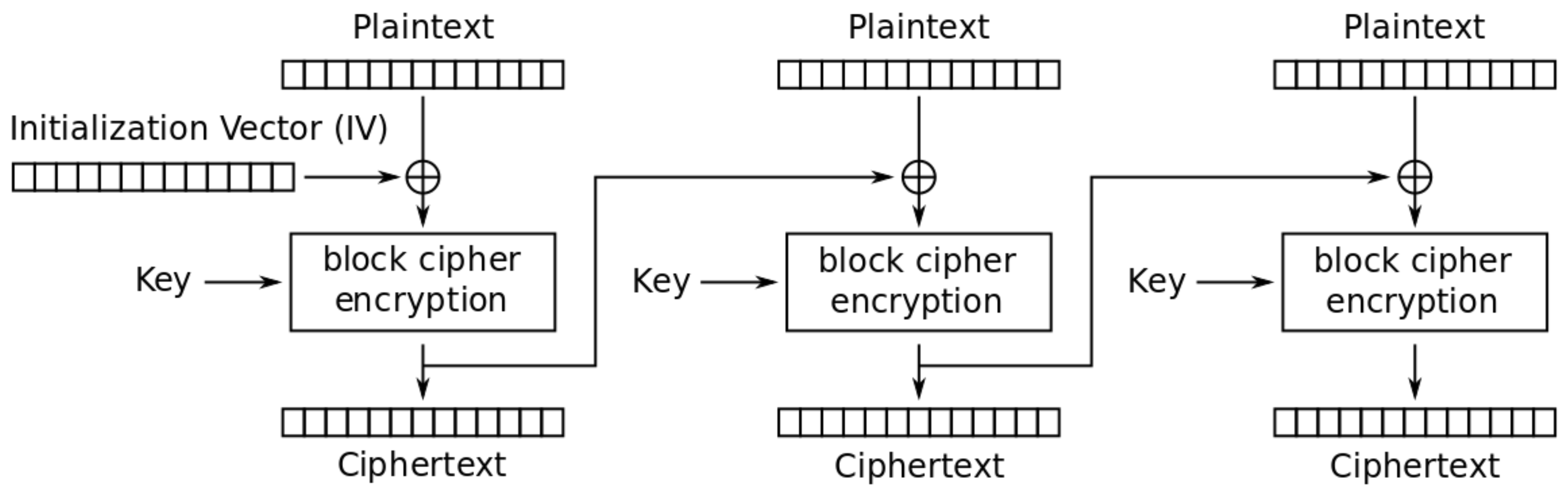


Electronic Codebook (ECB) mode encryption

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

- Glorified stream cipher with the speed benefits
- Does not use an IV
- Ciphertext blocks can contain repetition patterns
- Only good for small exchanges such as key transfers

Block Ciphers
CBC Mode

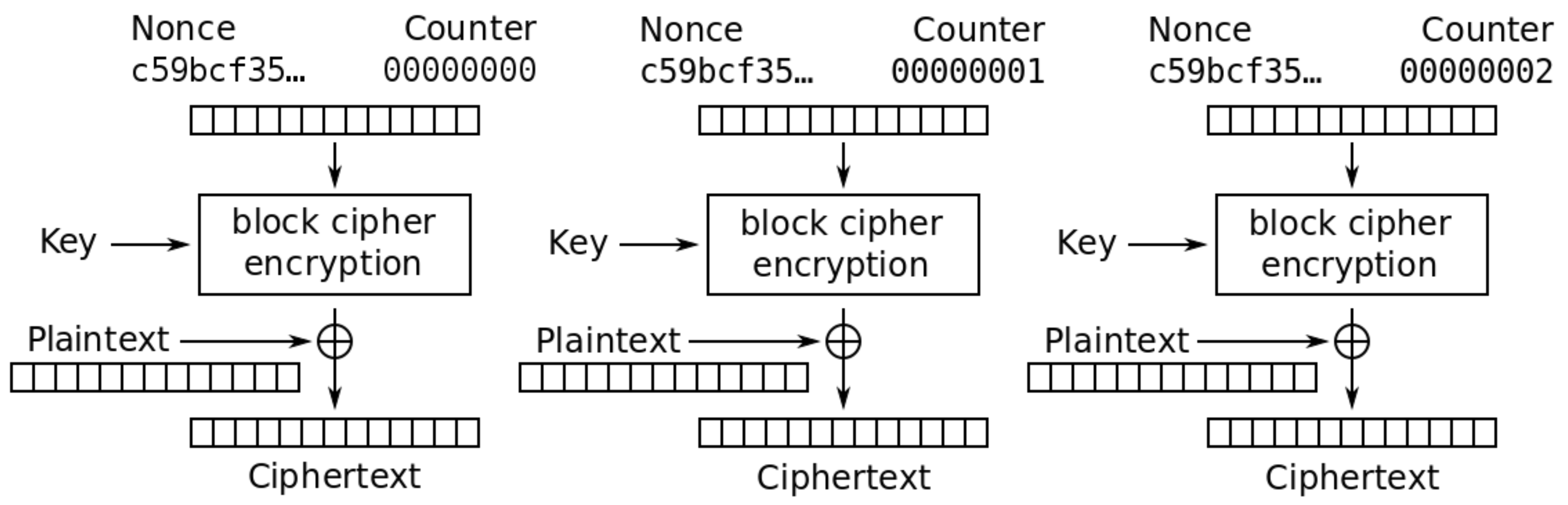


Cipher Block Chaining (CBC) mode encryption

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

- Pure block mode that requires padding input data to a multiple of the block size
- Cannot be parallelized
- First block is combined with an IV to create input to the cipher along with the key
- Each cipherblock is combined with the next plaintext block to create input to the cipher along with the key
- Can create patterns in ciphertext allowing deduction of the IV, and susceptible to padding-based attacks
- Cipher Feedback (CFB) and Output Feedback (OFB) modes are modified versions of CBC designed to avoid some of the weakness in CBC

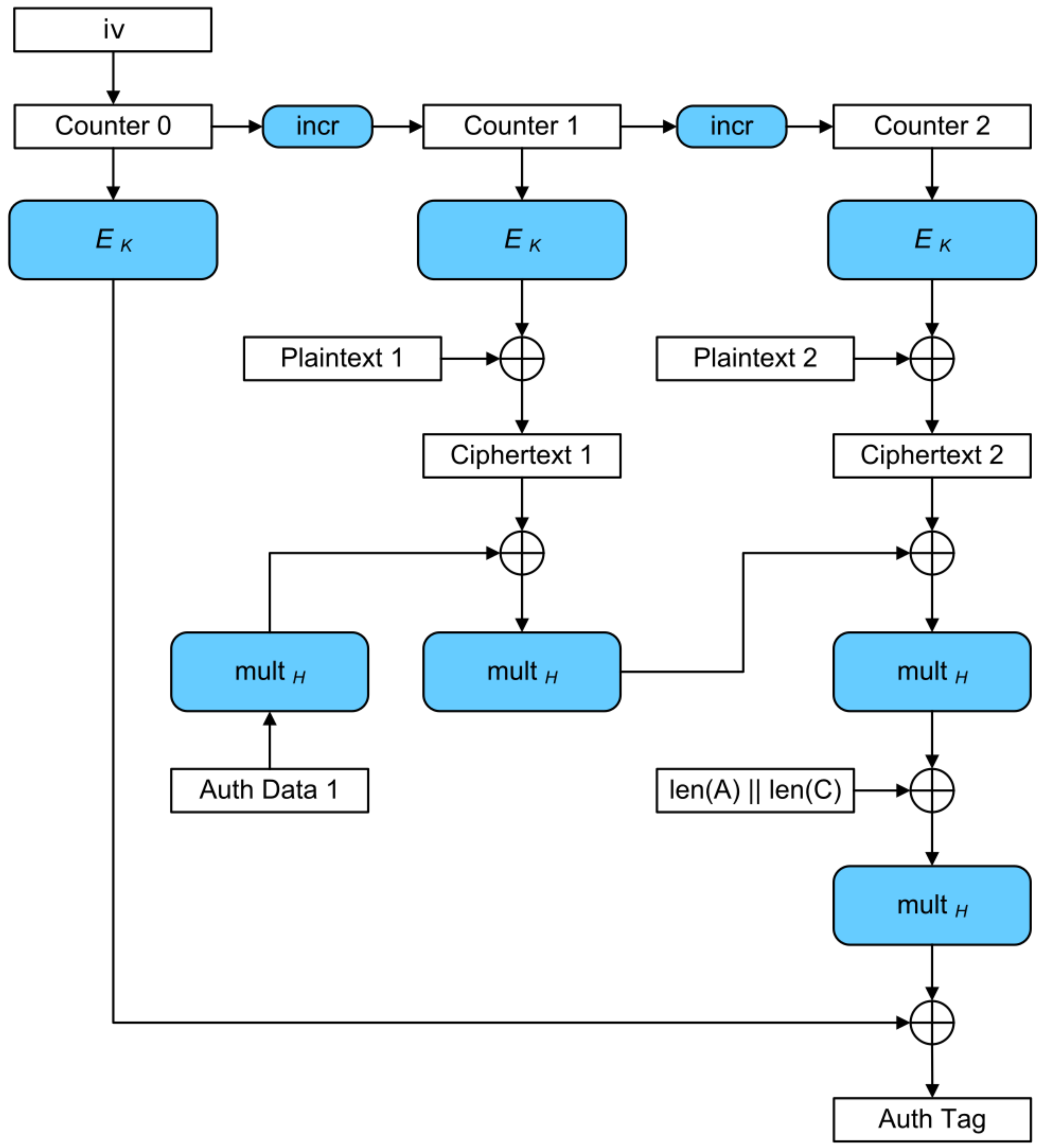
Block Ciphers
CTR Mode



Counter (CTR) mode encryption

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

- A streaming mode
- Suitable for parallelized processing, like ECB
- IV/Nonce may be random, or may come from any other number source
- Careful choices must be made in sources for the IV/Nonce, or chosen plaintext attacks are possible



https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Block Ciphers
GCM Mode

- A streaming mode
- Suitable for parallelized processing
- Generates authentication code for integrity as part of algorithm
- Galois Message Authentication Code (GMAC) is a way of using GCM to only produce the Auth Tag

Tools

https://www.cyberdefensemagazine.com/eset-crysis-decryptor-to-rescue-files-encrypted-by-the-crysis-ransomware/

aespipe

A cli tool to do AES encryption/decryption

```
cat /etc/hosts|file -

aespipe </etc/hosts -p 4 4<<<"1234567890abcdefghij"|file -

aespipe </etc/hosts -p 4 4<<<"1234567890abcdefghij"|hexdump

aespipe </etc/hosts -p 4 4<<<"1234567890abcdefghij"|
 aespipe -d -p 4 4<<<"1234567890abcdefghij" |
 file -
```

- Open source software from Jari Ruusu
- Source is available at http://loop-aes.sourceforge.net
- Can do single key, double key, or triple key
- Can integrate with GnuPG tools, keys, and ids
- Has many options

ccrypt
Another cli tool to do AES

```
ccencrypt -K Password01 </etc/hosts | file -

ccencrypt -K Password01 </etc/hosts | hexdump

ccencrypt -K Password01 </etc/hosts |
 ccdecrypt -K Password01
```

- ccrypt is a simple tool to just do AES-256
- ccrypt still has quite a few options
- ccrypt can be installed with apt
- ccrypt is primarily intended to be used to encrypt files in place

Openssl

Linux isn't the only platform for encryption

```
openssl aes-256-cbc </etc/hosts | file -

openssl aes-256-cbc </etc/hosts | hexdump

openssl aes-256-cbc </etc/hosts |
 openssl aes-256-cbc -d
```

- From cipher on Windows to openssl on MacOSX, all the popular operating systems have some free cli tools available
- Some are special-purpose like zip and ccrypt, other are swiss army knives like openssl
- There are graphical tools which tend to be aimed more at being toys or educational aids
- openssl is a good cross-platform tool to learn because of its ubiquity
- Web-based tools can enable exploration of less-common ciphers - e.g. dcode.fr

aescrypt

Cross-platform OSS tool with both command line and GUI versions

- www.aescrypt.com has the software and docs
- Intended to be used for file encryption, but can do encryption of streaming data in scripts too
- Runs on all the popular operating systems

Yet another variable

- Data encrypted with one of these tools is best decrypted using the same tool
  - sometimes they do not use the same methods of generating the keystream
  - sometimes they use different round counts
  - sometimes the docs are incorrect regarding the defaults for encryption parameters
- When decrypting, you need to know more than "it was encrypted with AES and this key"

Cipher Creation

DON'T

Your inability to imagine how to break your own creation is in no way a reflection of the abilities of others