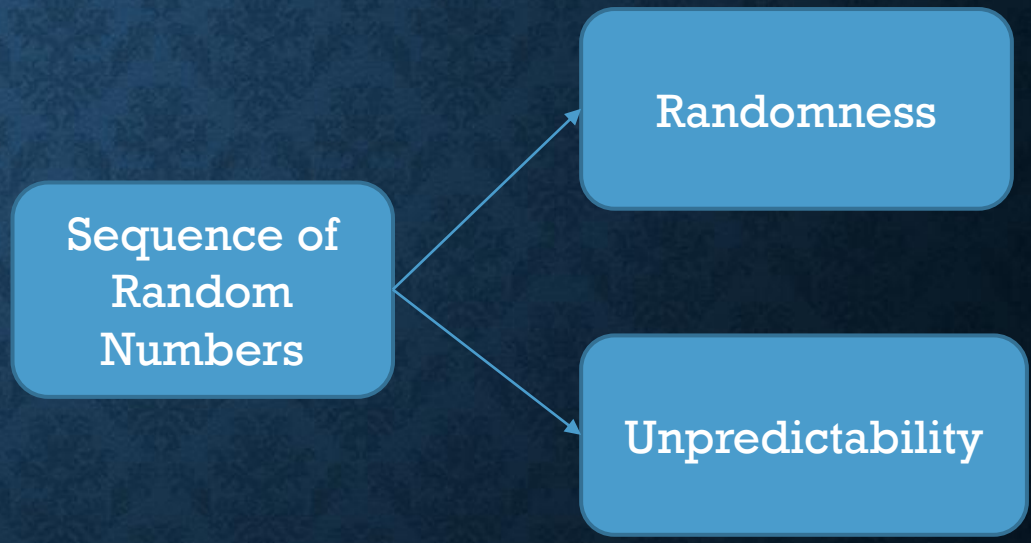
CRYPTOGRAPHIC RANDOM NUMBERS

RANDOM NUMBERS

- Most cryptographic algorithms and protocols require the use of a "random" key.
- Kerchoff's Principle dictates that the strength of the protocol depends on the size of the keysapce and the unpredicatibility of the key.
- Since speed is a critical factor, various cryptographic protocols make use of random binary numbers.



RANDOMNESS

Two criteria are used to validate that a sequence of numbers is random:

- Uniform distribution
  - The frequency of occurrence of ones and zeros is approximately equal
- Independence
  - No one subsequence in the sequence can be inferred from the others

UNPREDICTABILITY

- Successive numbers of a pseudorandom sequence are unpredictable.
- With “true” random sequences each number is statistically independent of other numbers in the sequence and therefore unpredictable

TRUE RANDOM NUMBER GENERATORS (TRNG)

- Take as input a source that is effectively random
- The source is referred to as an entropy source and is drawn from the physical environment of the computer
  - Includes things such as keystroke timing patterns, disk electrical activity, mouse movements, and instantaneous values of the system clock
  - The source, or combination of sources, serve as input to an algorithm that produces random binary output
- A TRNG may simply convert an analog source to a binary output
- A TRNG may involve additional processing to overcome any bias in the source

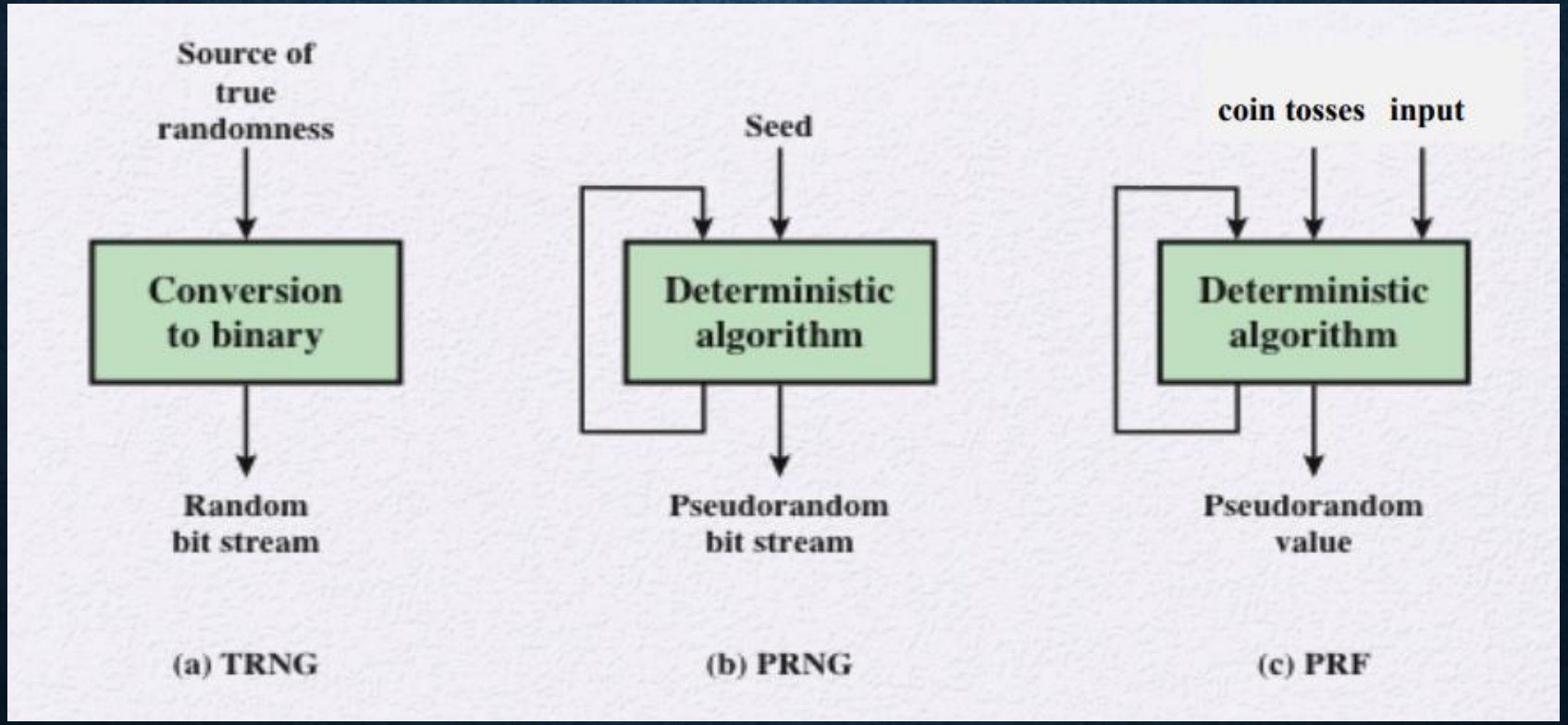
PSEUDORANDOM NUMBER GENERATORS (PRNG) AND PSEUDORANDOM FUNCTIONS(PRF)

- A PRNG takes as input a fixed value, called the seed, and produces a sequence of output bits using a deterministic algorithm
  - Quite often the seed is generated by a TRNG
  - The output bit stream is determined solely by the input, so an adversary who knows the algorithm and the seed can reproduce the entire bit stream
- A PRF takes as input a string of given length and outputs a pseudorandom sequence of fixed length. The inputs need not be random.

PSEUDORANDOM NUMBER GENERATORS (PRNG) AND PSEUDORANDOM FUNCTIONS(PRF)

- Pseudorandom number generator (PRNG)
  - An algorithm that is used to produce an open-ended sequence of bits
  - Input to a symmetric stream cipher is a common application
- Pseudorandom function (PRF)
  - Used to produce a pseudorandom string of bits of some fixed length
  - Examples are symmetric encryption keys and nonces

RNG'S AND RNF'S



PSEUDORANDOM NUMBER GENERATORS

- The generated bit stream needs to appear random even though it is deterministic
  - There is no single test that can determine if a PRNG generates numbers that have the characteristic of randomness
  - If the PRNG exhibits randomness on the basis of multiple tests, then it can be assumed to satisfy the randomness requirement

RANDOMNESS TESTS

- NIST SP 800-22 lists several tests to check randomness
- Frequency test
  - The number of ones and zeros in a sequence is approximately the same as would be expected for a truly random sequence
- Runs test
  - The total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits bounded before and after with a bit of the opposite value
  - The number of runs of ones and zeros of various lengths is as expected for a random sequence
- Maurer's universal statistical test
  - Find whether or not the sequence can be significantly compressed without loss of information. A significantly compressible sequence indicates non-randomness

UNPREDICTABILITY

- Forward unpredictability
  - The next output bit in the sequence should be unpredictable, in spite of any knowledge of previous bits in the sequence
- Backward unpredictability
  - It should not be feasible to determine the seed from knowledge of any generated values.
- The same set of tests for randomness also provides a test of unpredictability
- A random sequence will have no correlation with a fixed value (the seed)

SEED REQUIREMENTS

- The seed itself must be a random or pseudorandom number
- Typically, the seed is generated by a TRNG

ALGORITHM DESIGN

Three broad categories of cryptographic algorithms are commonly used to create PRNGs:

- Symmetric block ciphers
- Asymmetric ciphers
- Hash functions and message authentication codes

LINEAR CONGRUENTIAL GENERATOR

The sequence of positive integers obtained by

$$X_{n+1} = (aX_n + c) \bmod m$$

where m , a , c , and X0 are integers with:

m the modulus $m > 0$

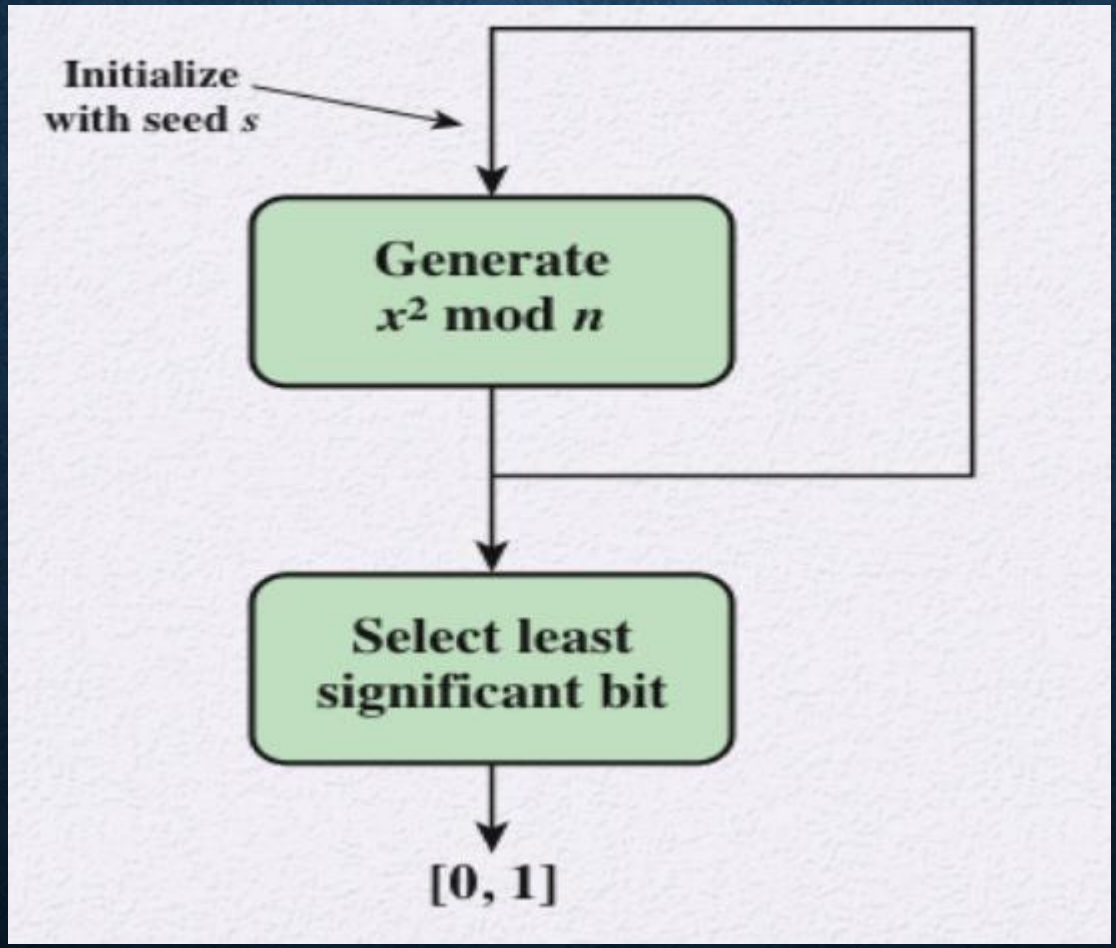
a the multiplier $0 < a < m$

c the increment $0 \leq c < m$

X_0 the starting value, or seed

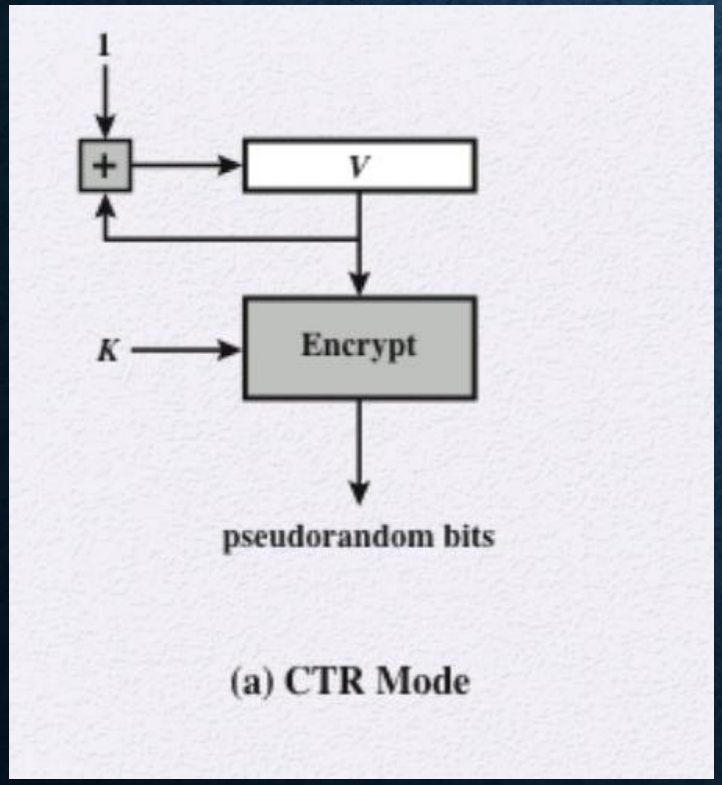
$0 \leq X_0 < m$ is pseudorandom if the values for a , c , and m are selected appropriately.

BLUM BLUM SHUB GENERATOR (BBS)

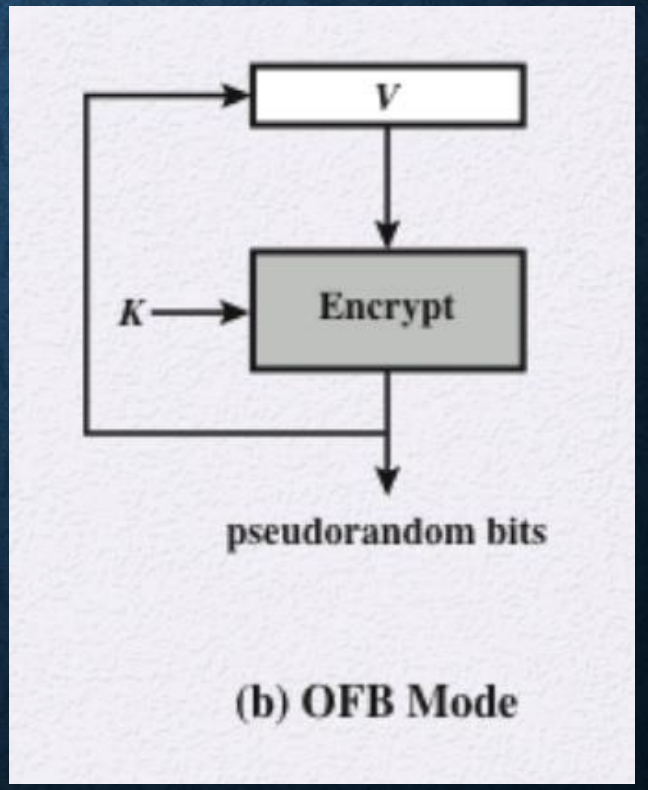


PRNG USING BLOCK CIPHER MODES OF OPERATION

- CTR Mode (counter)



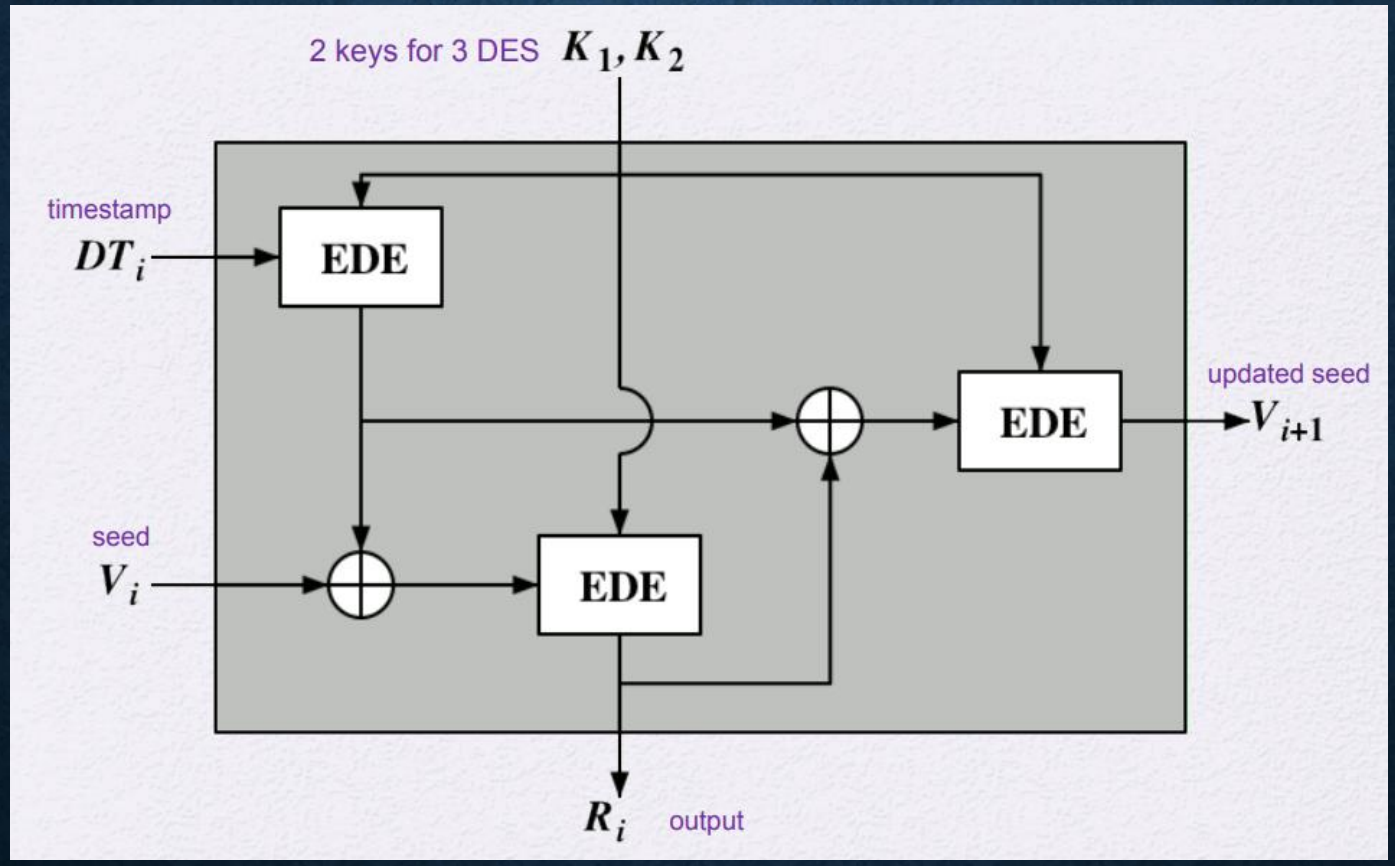
- OFB Mode (Output Feedback)



ANSI X9.17 PRNG

- A number of applications use this, including financial security applications and PGP
- Two inputs
  - a 64-bit representation of the current date and time (timestamp)
  - a 64-bit seed value that is initialized to some arbitrary value and is updated during the generation process.
- Keys
  - The generator makes use of three 3 DES encryption.
  - All three make use of the same pair of 56-bit keys.
- Output
  - a 64-bit pseudorandom number
  - a 64-bit seed value.

ANSI X9.17 PRNG



ENTROPY SOURCES

- A true random number generator (TRNG) uses a nondeterministic source to produce randomness
- Most operate by measuring unpredictable natural processes such as pulse detectors of ionizing radiation events, gas discharge tubes, and leaky capacitors
- Intel has developed a commercially available chip that samples thermal noise by amplifying the voltage measured across undriven resistors
- LavaRnd is an open source project for creating truly random numbers using inexpensive cameras, open source code, and inexpensive hardware

POSSIBLE SOURCES OF RANDOMNESS

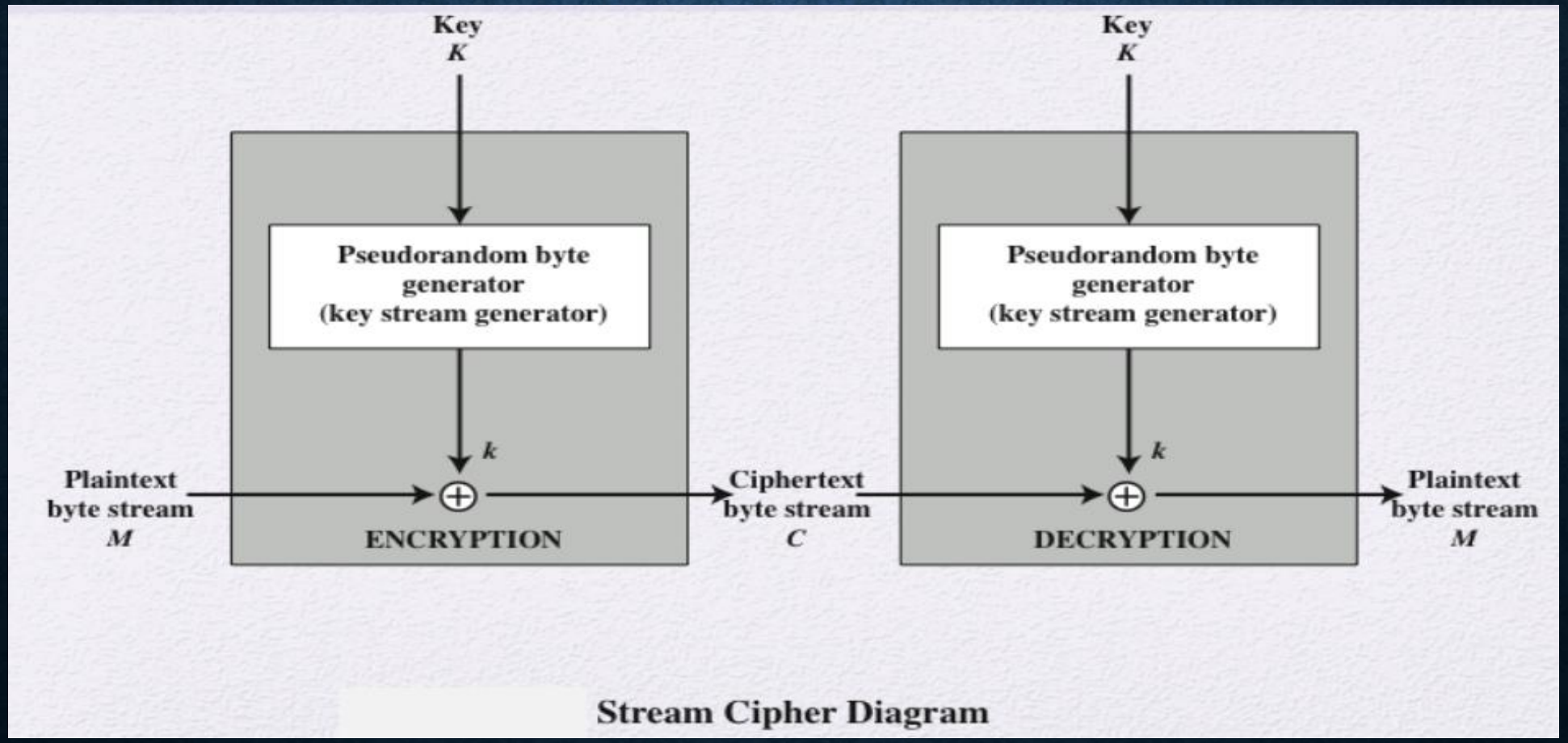
- Sound/video input
  - The input from a sound digitizer with no source plugged in, a camera with the lens cap on
  - If the system has enough gain to detect anything, such input can provide reasonable high quality random bits
- Disk drives
  - Have small random fluctuations in their rotational speed due to chaotic air turbulence
  - The addition of low-level disk seektime instrumentation produces a series of measurements that contain this randomness

SKEW

A TRNG may produce an output that is biased in some way, such as having more ones than zeros or vice versa

- Deskewing algorithms
  - Methods of modifying a bit stream to reduce or eliminate the bias
  - One approach is to pass the bit stream through a hash function such as MD5 or SHA-1
  - RFC 4086 recommends collecting input from multiple hardware sources and then mixing these using a hash function to produce random output
- Operating systems typically provide a built-in mechanism for generating random numbers
  - Linux uses four entropy sources: mouse and keyboard activity, disk I/O operations, and specific interrupts
  - Bits are generated from these four sources and combined in a pooled buffer
  - When random bits are needed the appropriate number of bits are read from the buffer and passed through the SHA-1 hash function

STREAM CIPHERS



Stream Cipher Diagram

STREAM CIPHER DESIGN

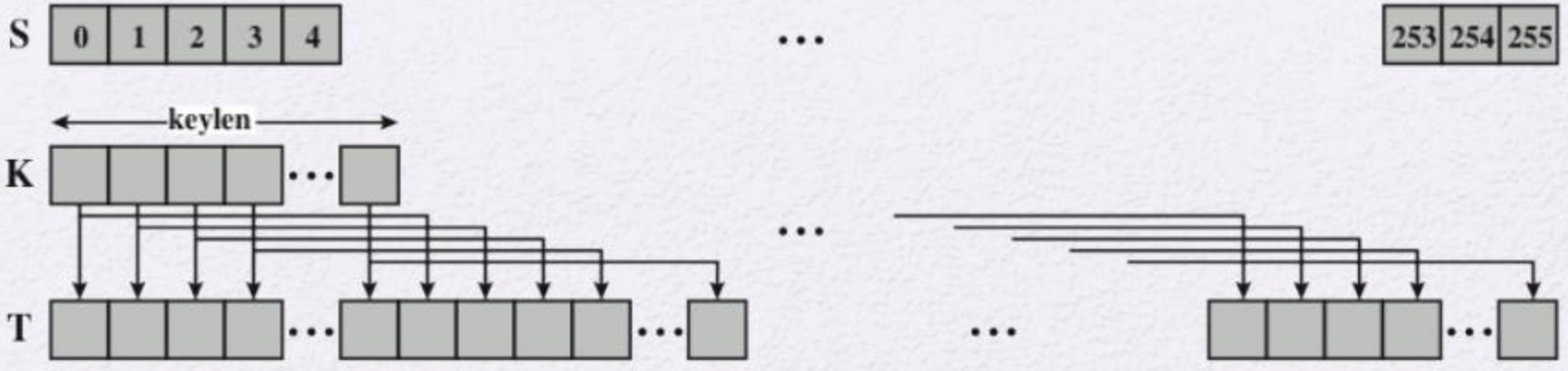
- The encryption sequence should have a large period
  - A PRNG produces a deterministic stream of bits that eventually repeats; the period should be large.
- The keystream should approximate the properties of a true random number stream
  - There should be an approximately equal number of 1s and 0s
  - If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear equally often

STREAM CIPHER DESIGN

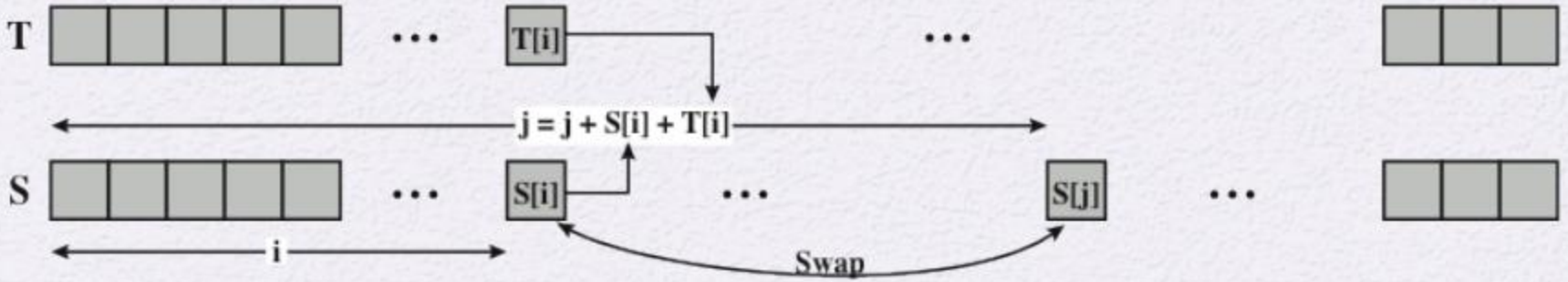
- Key length of at least 128 bits
  - The output of the PRNG depends on the value of the key
- With a properly designed PRNG a stream cipher can be as secure as a block cipher of comparable key length
  - Stream ciphers that do not use block ciphers as a building block are typically faster and use far less code than block ciphers

RC4

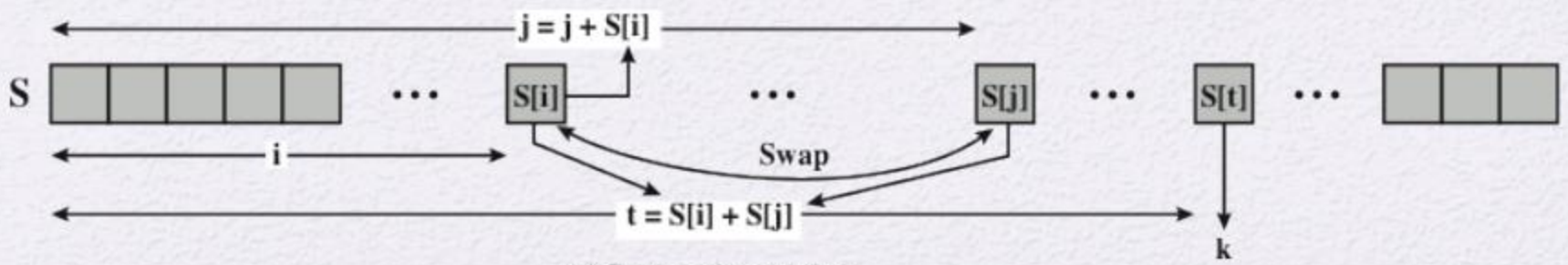
- Designed in 1987 by Ron Rivest for RSA Security
- Variable key size stream cipher with byte-oriented operations
- Based on the use of a random permutation
- Eight to sixteen machine operations are required per output byte and the cipher can be expected to run very quickly in software
- Used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards for communication between Web browsers and servers
- Used in the Wired Equivalent Privacy (WEP) protocol and the newer WiFi Protected Access (WPA) protocol (IEEE 802.11 wireless LAN standard)



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

STRENGTH OF RC4

- A number of papers have been published analyzing methods of attacking RC4
  - None of these approaches is practical against RC4 with a reasonable key length
- A more serious problem is that the WEP protocol intended to provide confidentiality on 802.11 wireless LAN networks is vulnerable to a particular attack approach
  - The problem is not with RC4 itself, but the way in which keys are generated for use as input
  - Problem does not appear to be relevant to other applications and can be remedied in WEP by changing the way in which keys are generated
  - Problem points out the difficulty in designing a secure system that involves both cryptographic functions and protocols that make use of them