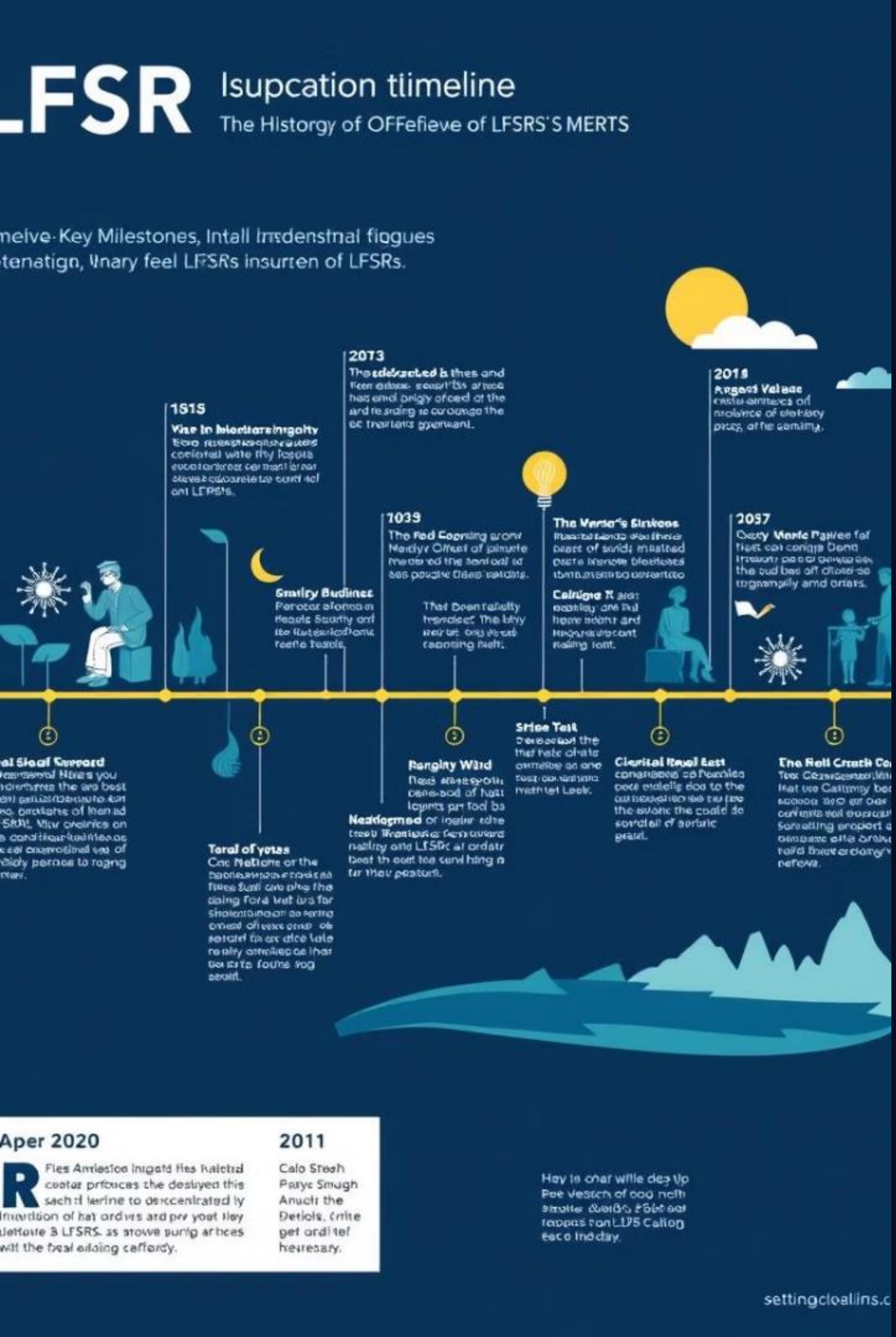
LFSR Method in Cryptography

The LFSR (Linear Feedback Shift Register) is a fundamental cryptographic algorithm used to generate pseudorandom sequences of bits. It is widely employed in various encryption techniques to enhance the security and randomness of data transmission.

by Sajjad Ranjbar



History and Background of LFSR

1

1950s

LFSRs were initially developed by Claude Shannon and others as a way to generate pseudorandom sequences for cryptographic applications.

2

1960s-70s

LFSRs were widely adopted in various communication systems, including stream ciphers, digital communications, and error-correcting codes.

3

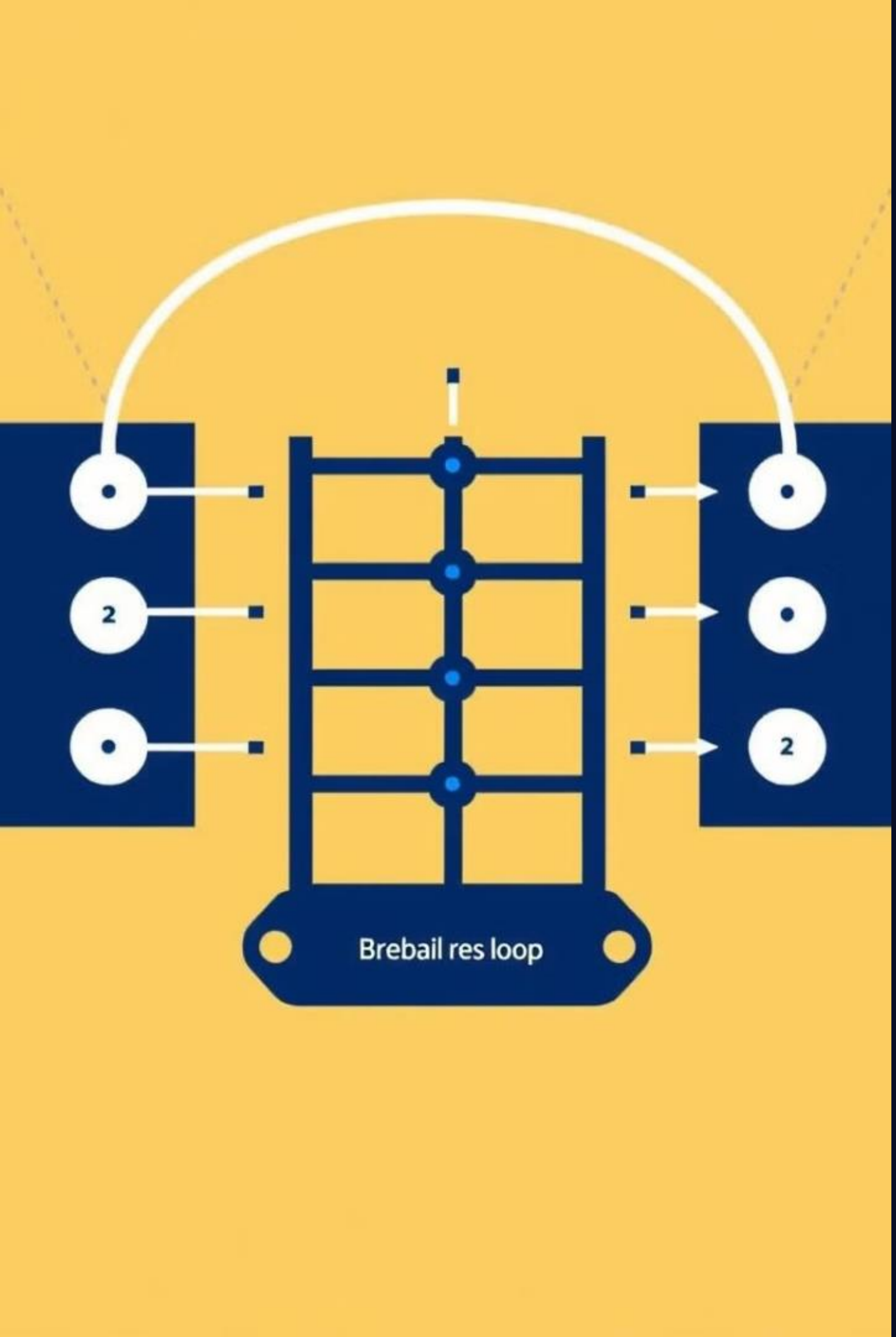
Present Day

LFSRs continue to be a crucial component in modern cryptographic algorithms, ensuring the security and integrity of data transmission.

Theorem: *The maximum sequence length generated by an LFSR of degree m is* $2^m - 1$.

Theorem: *The maximum sequence length generated by an LFSR of degree m is* $2^m - 1$.

It is easy to show that this theorem holds. The *state* of an LFSR is uniquely determined by the m internal register bits. Given a certain state, the LFSR deterministically assumes its next state. Because of this, as soon as an LFSR assumes a previous state, it starts to repeat. Since an m-bit state vector can only assume $2^m - 1$ nonzero states, the maximum sequence length before repetition is $2^m - 1$



Principles of the LFSR Algorithm

Shift Register

The LFSR algorithm utilizes a shift register, where a sequence of bits is stored and shifted to the right with each iteration.

Feedback Function

The feedback function determines the next bit in the sequence based on the current state of the shift register.

Pseudorandom Sequence

The LFSR generates a repeating, pseudorandom sequence of bits that appears random but can be reproduced with the same initial state.

Key Components of an LFSR

Key Components of an LFSR

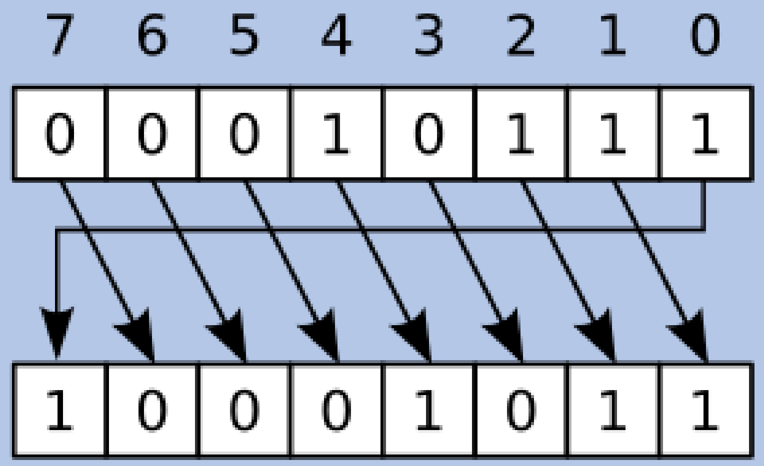
Shift Register

A series of flip-flops or memory cells that store the current state of the LFSR.

Key Components of an LFSR

Shift Register

A series of flip-flops or memory cells that store the current state of the LFSR.



Key Components of an LFSR

Shift Register

A series of flip-flops or memory cells that store the current state of the LFSR.

Feedback Function

A Boolean function that determines the next bit in the sequence based on the current state.

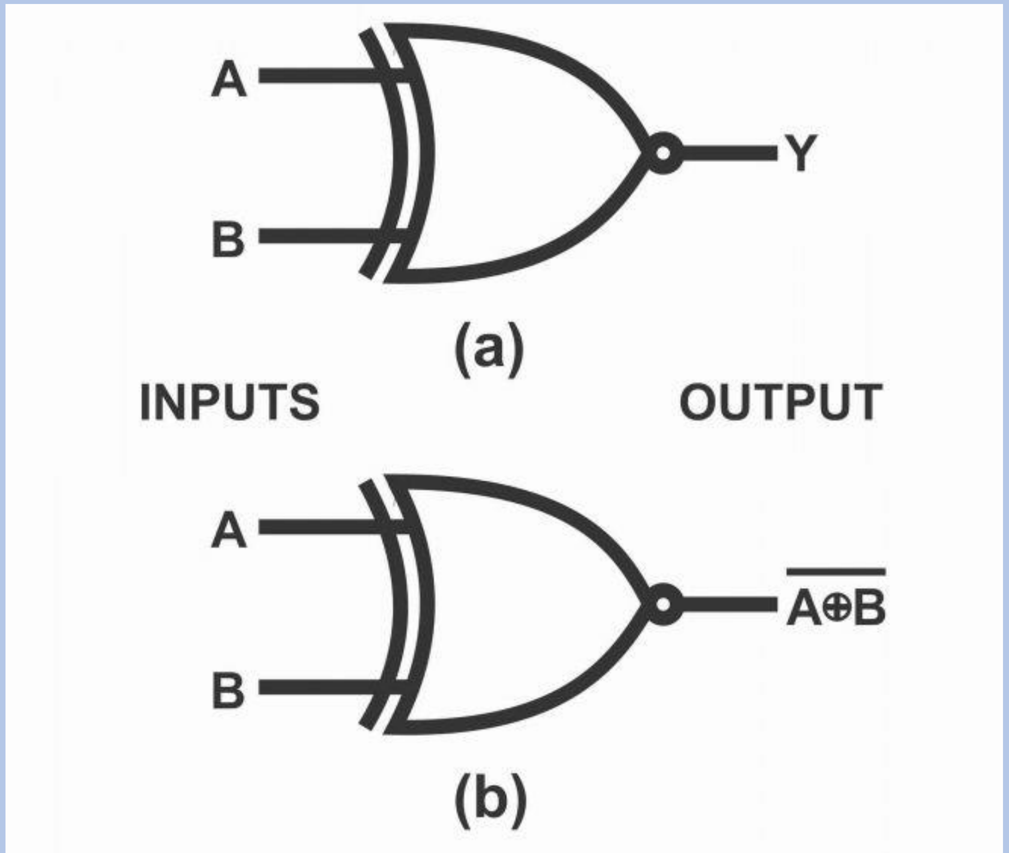
Key Components of an LFSR

Shift Register

A series of flip-flops or memory cells that store the current state of the LFSR.

Feedback Function

A Boolean function that determines the next bit in the sequence based on the current state.



Key Components of an LFSR

Shift Register

A series of flip-flops or memory cells that store the current state of the LFSR.

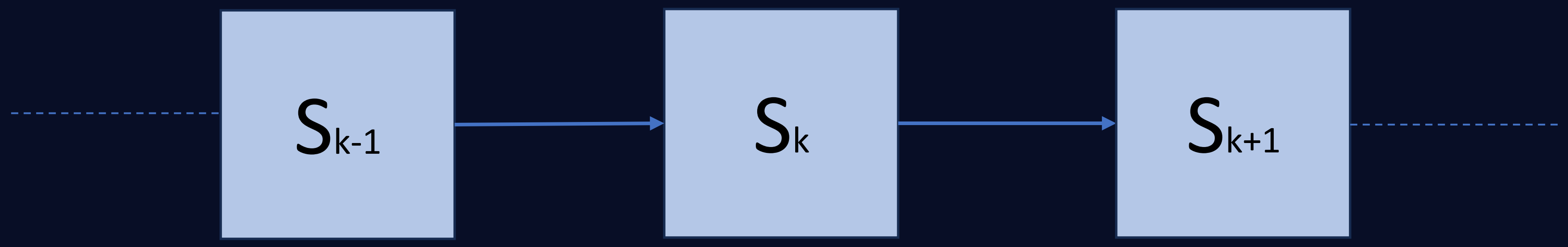
Feedback Function

A Boolean function that determines the next bit in the sequence based on the current state.

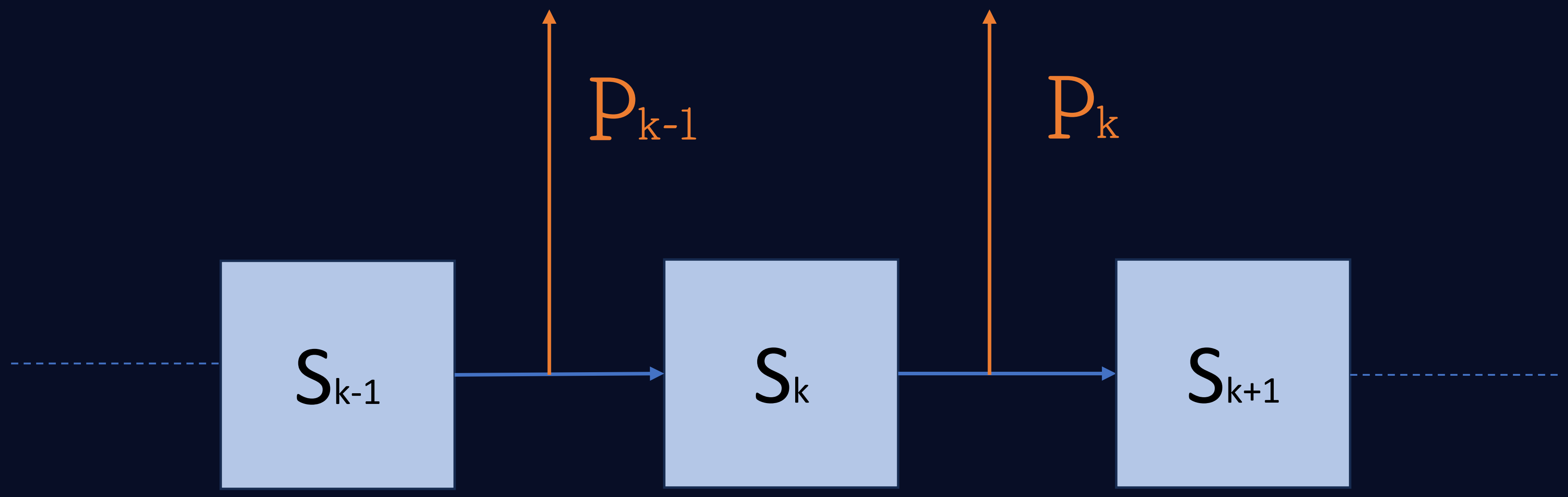
Taps

The specific bit positions in the shift register that are used in the feedback function.

Taps

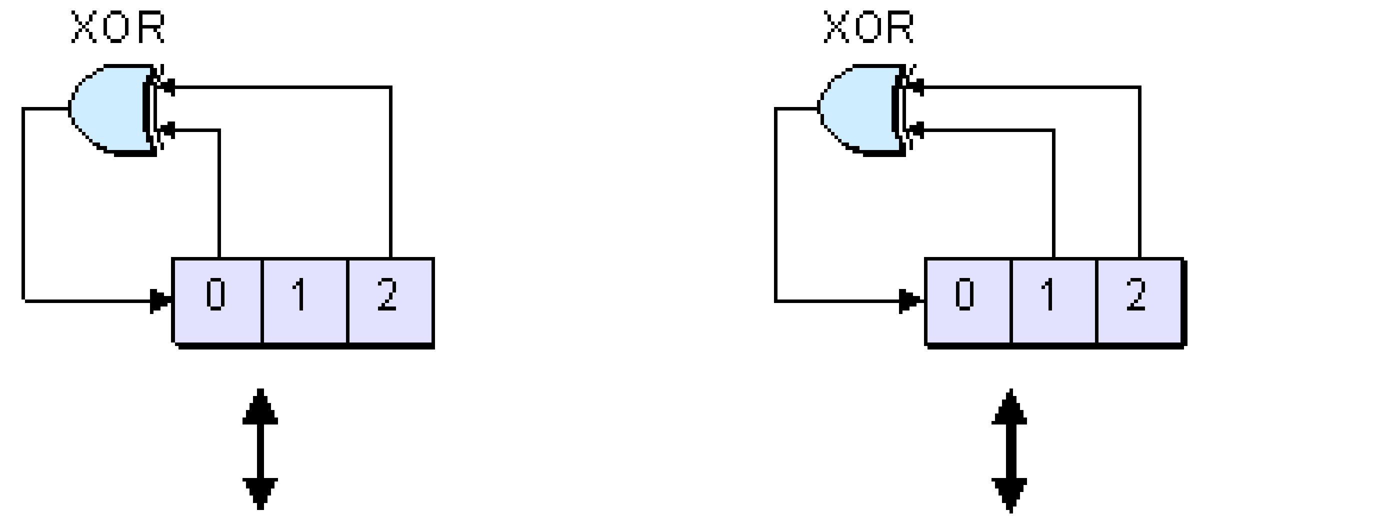


Taps



feedback coefficient

Taps

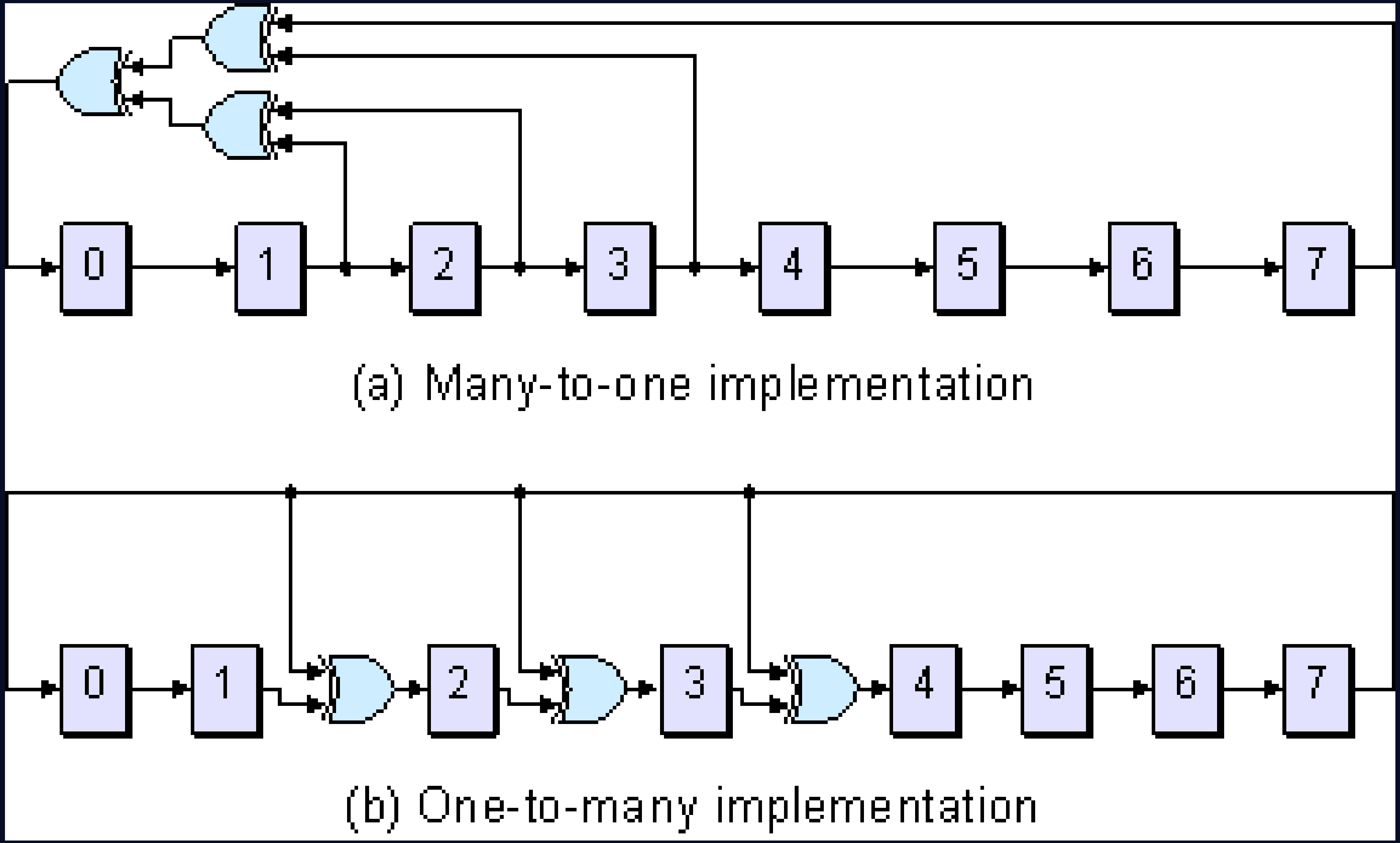


Clock	q0	q1	q2	
--	0	0	1	Initial value
1	1	0	0	
2	1	1	0	
3	1	1	1	
4	0	1	1	
5	1	0	1	
6	0	1	0	
7	0	0	1	
8	1	0	0	
9	1	1	0	
:	:	:	:	

Clock	q0	q1	q2	
--	0	0	1	Initial value
1	1	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	0	
5	1	1	1	
6	0	1	1	
7	0	0	1	
8	1	0	0	
9	0	1	0	
:	:	:	:	

Comparison of alternative tap selections

Taps



One-to-many versus many-to-one implementations

One to many *implementations*

One to many *implementations*

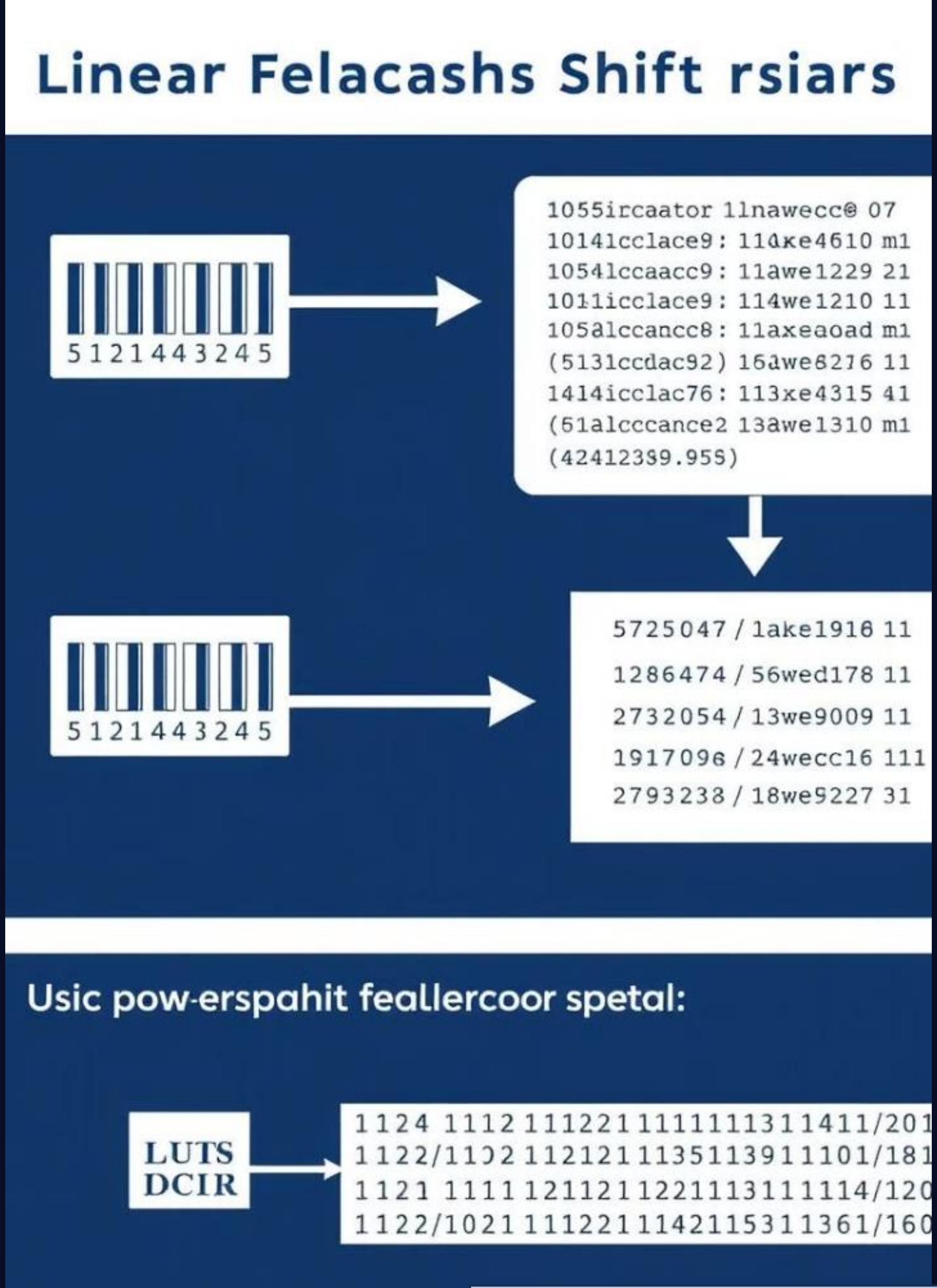
- If $p_i = 1$ (closed switch), the feedback is active.

One to many *implementations*

If p_i = 1 (closed switch), the feedback is active.

```
Successor(List ,taps)
{
  next  = shift. List
  for t in taps
    { List.rare = rare XOR t }
  return(next)
}
```

Solving an LFSR Example



LFSR Method in Cryptography

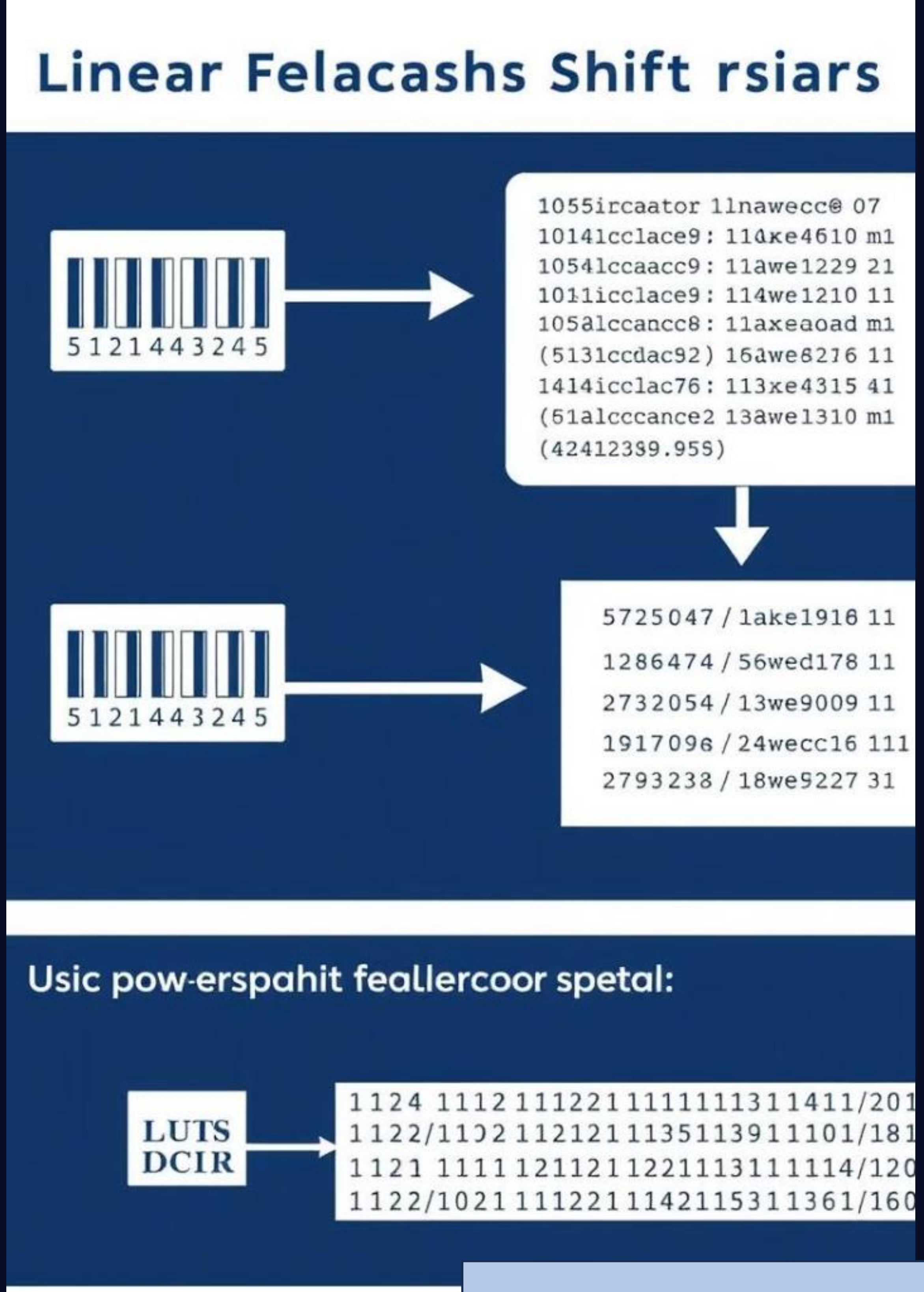
Solving an LFSR Example

1

Initial State

The register is initialized with the value 10100 taps={1,4}.

1	0	1	0	0

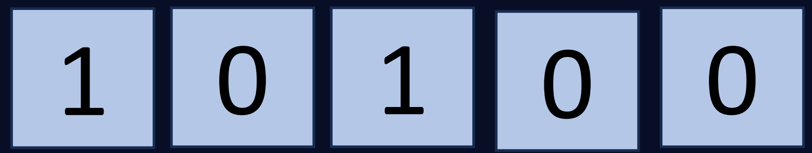


LFSR Method in Cryptography

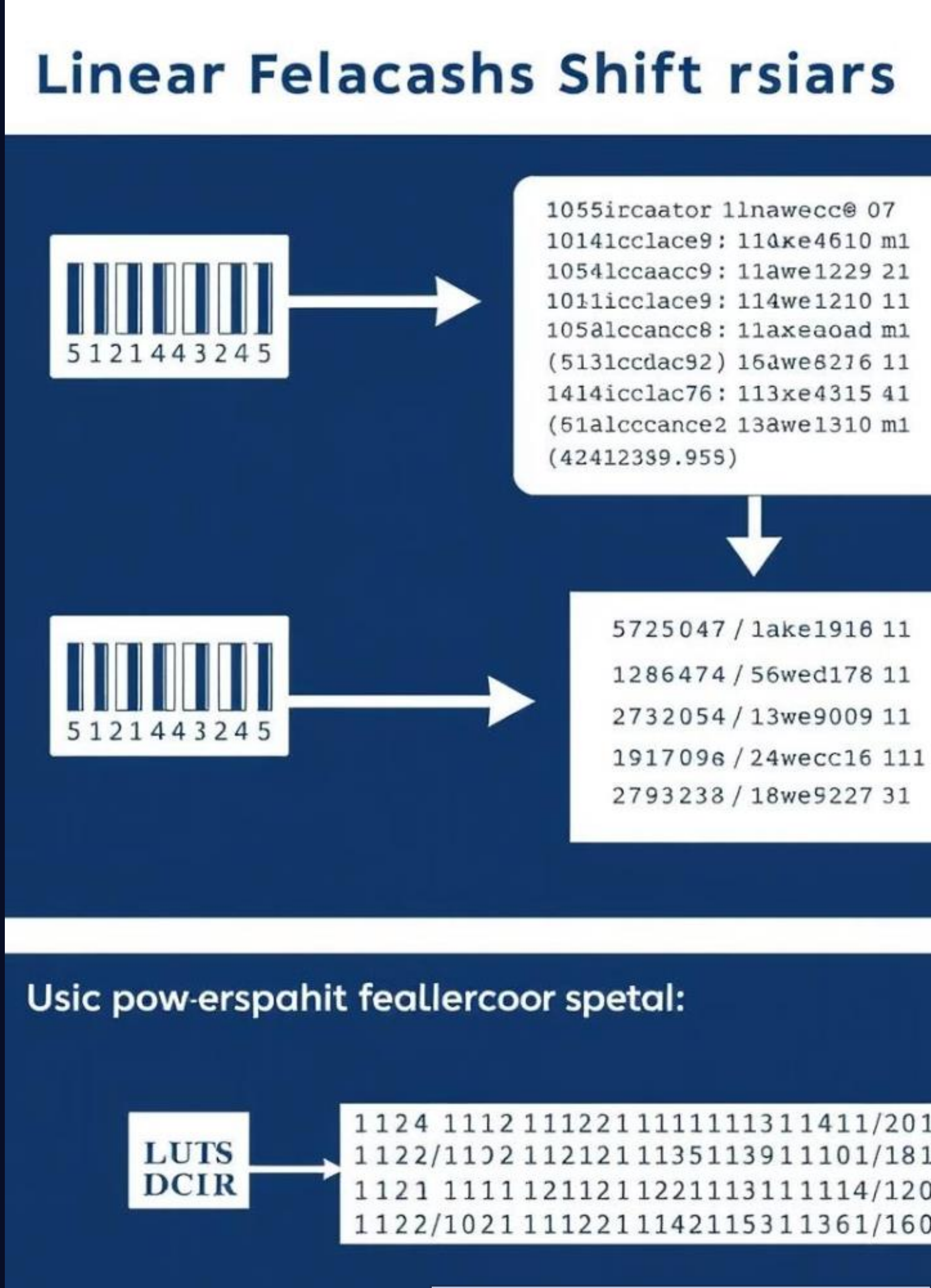
Solving an LFSR Example

1 Initial State

The register is initialized with the value 10100 taps={1,4}.



2 Shift and Feedback



LFSR Method in Cryptography

Solving an LFSR Example

1. Initial State

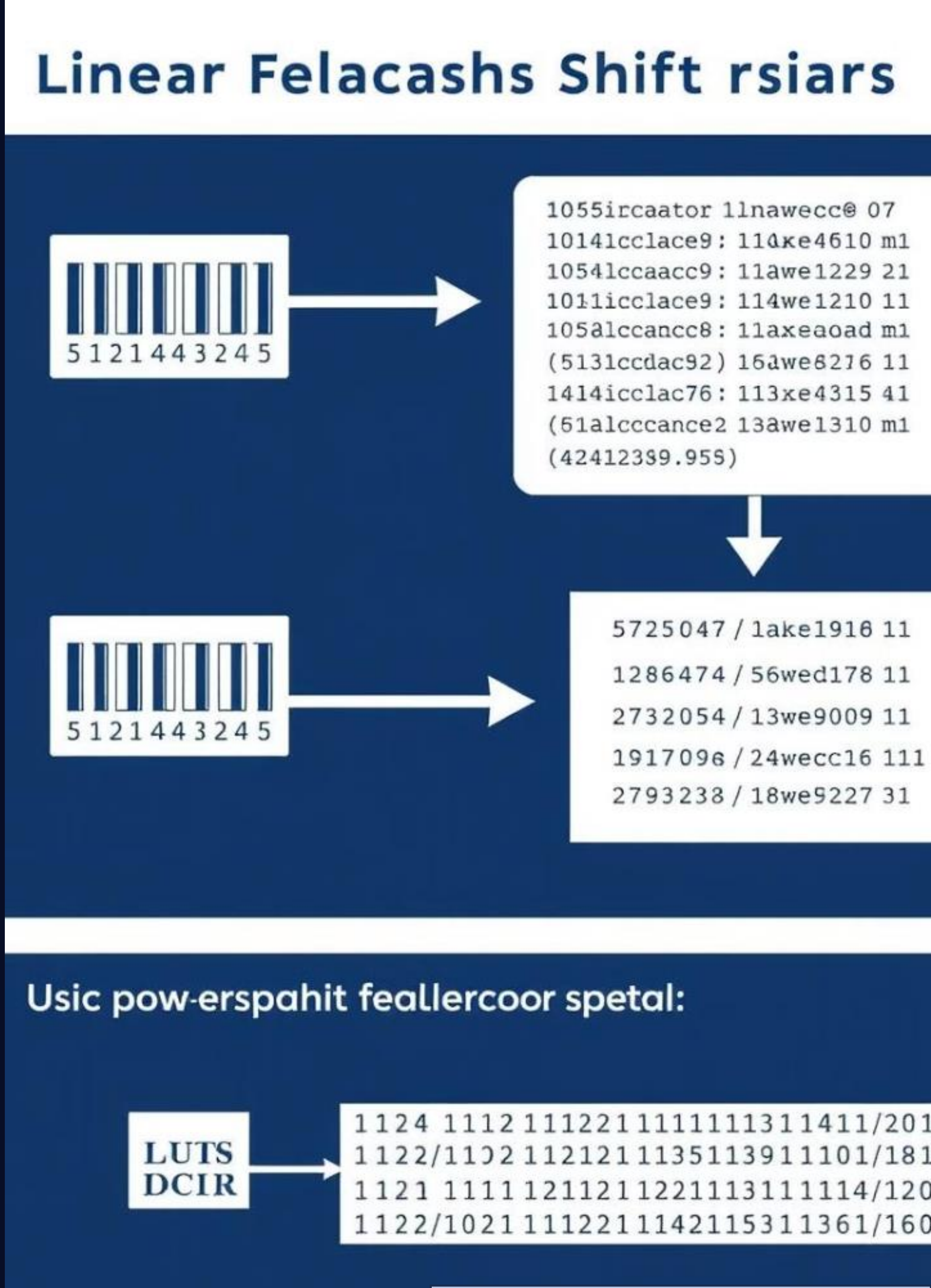
The register is initialized with the value 10100 taps={1,4}.

1	0	1	0	0

2. Shift and Feedback

1	0	1	0	0

((0 XOR 0) XOR 1) = 1



LFSR Method in Cryptography

Solving an LFSR Example

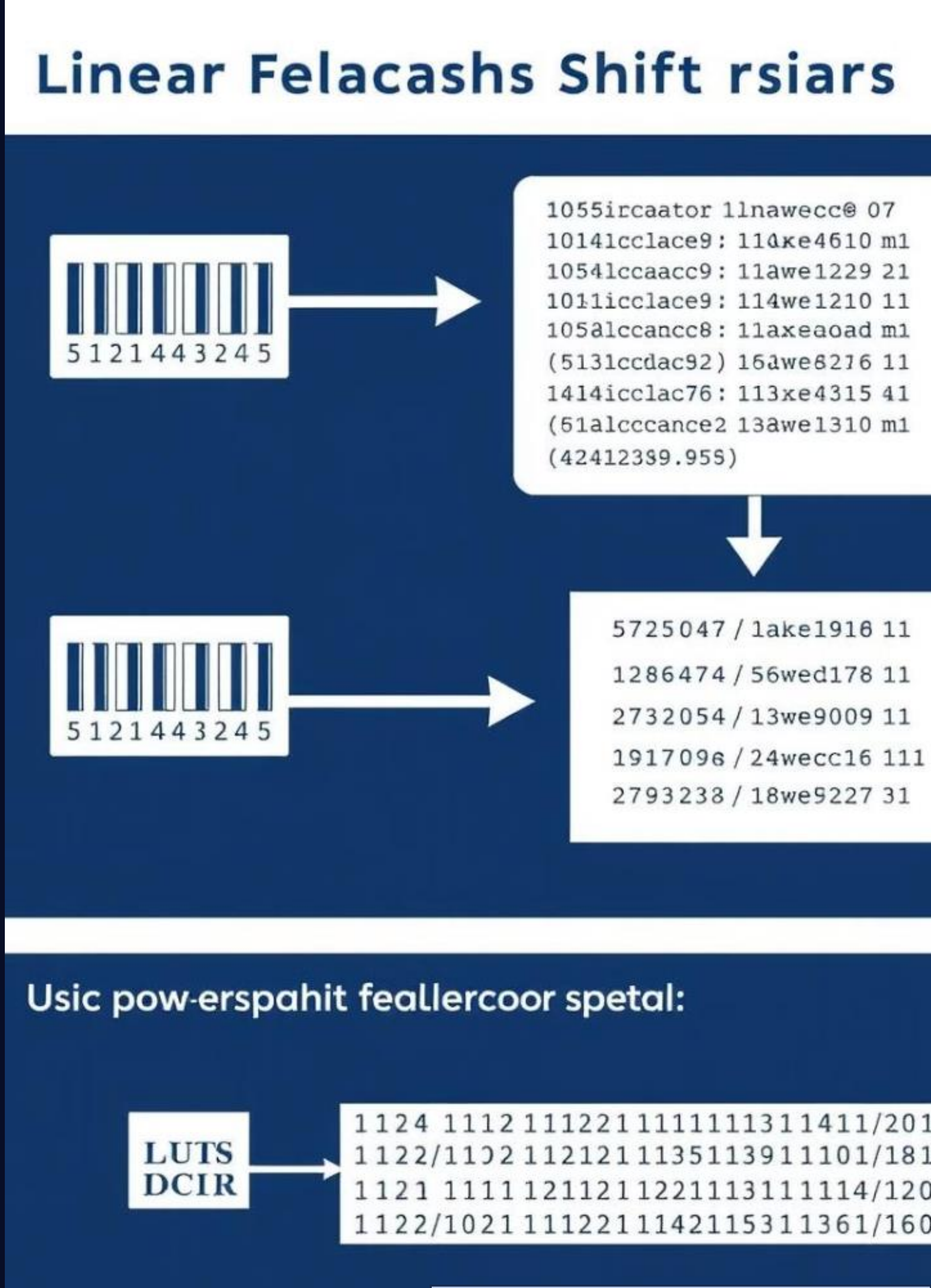
1. **Initial State**

The register is initialized with the value 10100 taps={1,4}.

1	0	1	0	0

2. **Shift and Feedback**

1	1	0	1	0



LFSR Method in Cryptography

Solving an LFSR Example

1

Initial State

The register is initialized with the value 10100 taps={1,4}.

1	0	1	0	0

2

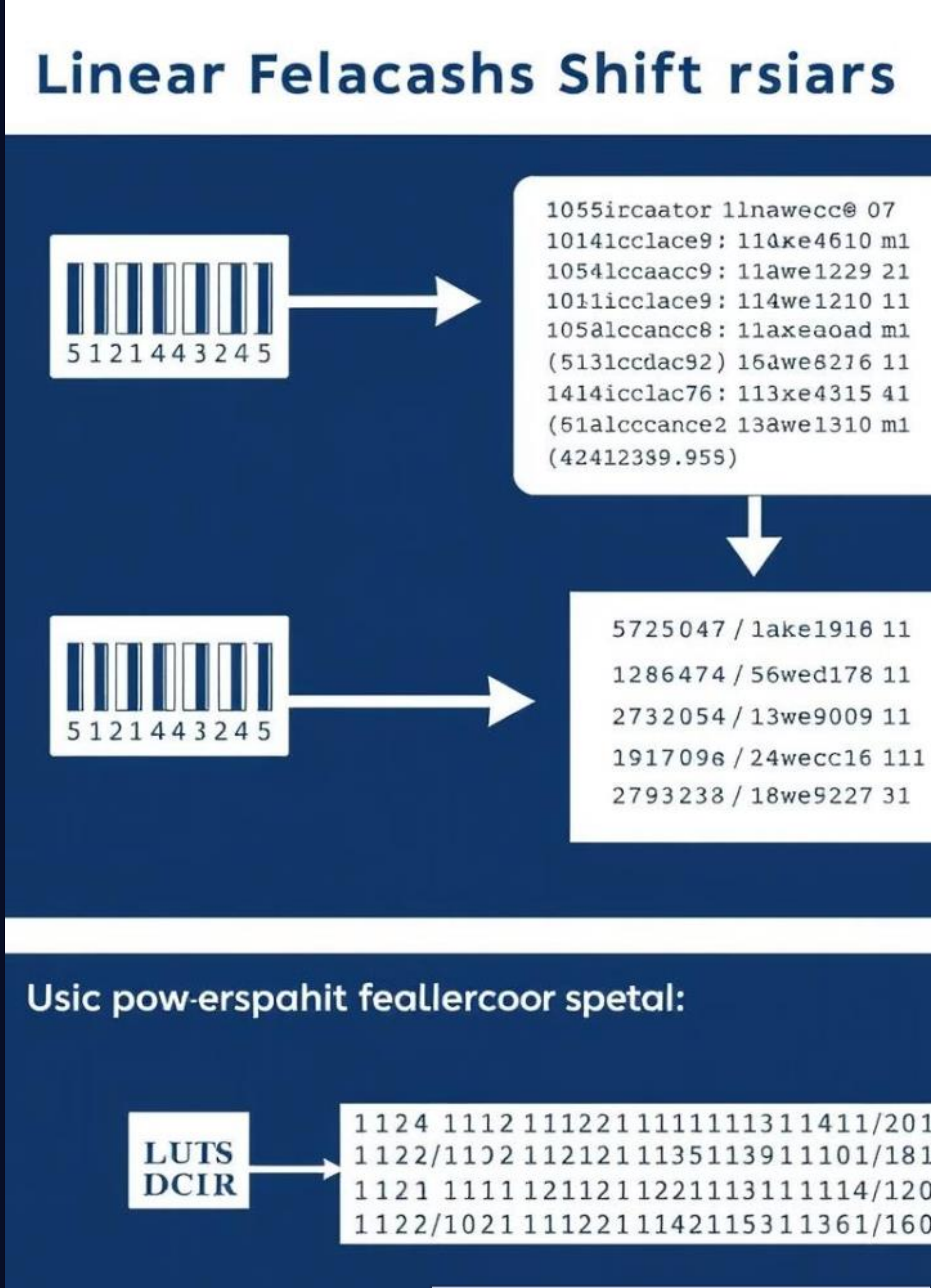
Shift and Feedback

1	1	0	1	0

3

Pseudorandom Sequence

The output of the LFSR generates a repeating pseudorandom sequence: 10100, 11010, ...



LFSR Method in Cryptography

Applications of LFSR in Cryptography

Stream Ciphers

LFSRs are widely used in stream ciphers to generate key streams for encrypting and decrypting data.

Pseudo-Random Number Generators

LFSRs are the foundation for many pseudo-random number generators (PRNGs) used in cryptographic applications.

Error-Correcting Codes

LFSRs are employed in the design of error-correcting codes to detect and correct transmission errors.

Conclusion and Future Considerations

1 Continued Importance

The LFSR algorithm remains a fundamental component in modern cryptography, ensuring the security and integrity of data transmission.

2 Advancements

Ongoing research explores ways to enhance the security and efficiency of LFSR-based cryptographic systems, adapting to emerging threats and technologies.

3 Future Trends

LFSR-based techniques are likely to continue playing a crucial role in the development of next-generation cryptographic solutions.

Thak you so much