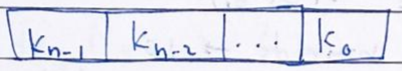
II) if the polynomial is irreducible then period of LFSR will divide $2^n - 1$

III) If it is reducible then different state will have different cycle length (different period)

n-bit LFSR



$k = (k_0, \ldots, k_{n-1})$

output bits x_i → keystream bits z_i

$m_i \oplus z_i = c_i$ → ciphertext bits

1) Known plaintext attack

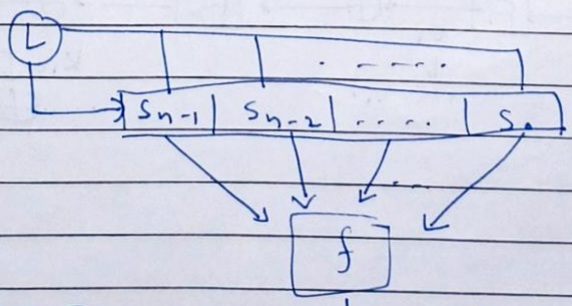
$z_0 = k_0, \; z_1 = k_1, \; \ldots, \; z_{n-1} = k_{n-1}$

if you have the keystream bits then you can prepare a system linear eqn.

2) LFSR with non-linear filter fn

$f: \{0,1\}^l \rightarrow \{0,1\}$

n-bit LFSR, $n \geq l$



state update of LFSR
1) linear feedback
2) shifting

$c_i = m_i \oplus z_i$

__/__/____

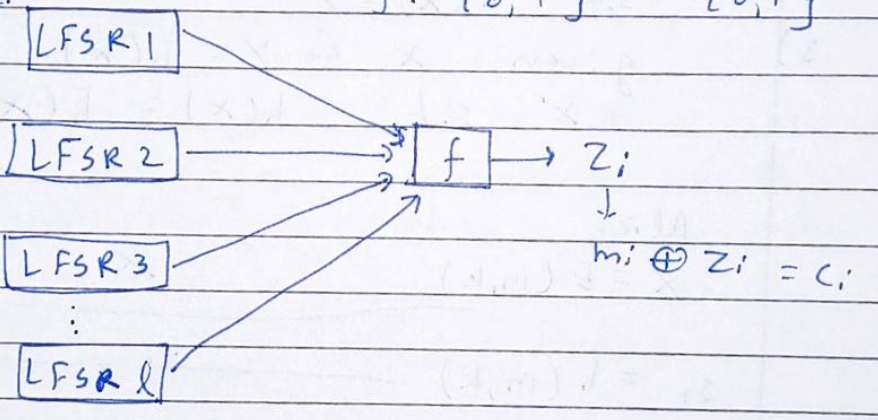
State update fn of LFSR is α

$$S_{t+1} = \alpha(S_t)$$

$$Z_{t+1} = f(S_{t+1})$$

(II) LFSR with combiner fn

$f: \{0,1\}^l \to \{0,1\}$



(III) non-linear feedback shift register (NFSR)

→ feedback fn is non-linear

$f: \{0,1\}^l \to \{0,1\}$

$t = f(x) + f(y) + f(x+y)$

$t = 0, 1$ | $f(x_0, x_1, x_2) = x_0 + x_1 x_2$

Hash Functions

$h: A \rightarrow B$

$h(x) = y$

1) if x is altered to x' then $h(x')$ will be completely different from $h(x)$
2) Given y you can't possibly find x s.t $h(x) = y$
3) given x & $y = h(x)$ you can't find x' s.t $h(x) = h(x')$

Alice

$x = E(m, k)$

$s_1 = h(m, k)$

$x \rightarrow$

$s_1 \rightarrow$

Bob

$x_1 = Dec(x_1, k)$

s_2

if $h(x_1, k) = s_2$ then Bob accepts x_1

* we are able to authenticate both the data & the source.

Defi^{n}: a hash fx is a four tuple (P, S, K, H) where:

1) P is set of all possible messages
2) S: set of all possible message digests or auth^{n} tags
3) K is the key space
4) for each $k_1 \in K$ there is a hash fx $h_{k_1} \in H$ s.t

$h_{k_1}: P \rightarrow S$ here $|P| \geq |S|$

more intresingly $|P| \geq 2 \times |S|$

- if key is involved in computation then it's called keyed hash fn, if not then unkeyed hash fn

Problem 1 (Preimage finding Problem)

$h: P \to S$

given $y \in S$ find $x \in P$ such that $h(x) = y$

if you can't find x in resonable time then h is preimage resistant hash fn, else it is preimage friendly.

Problem 2 (2^{nd} preimage finding problem)

$h: P \to S$

$x \in P$ & $h(x)$ find $x' \in P$ s.t $x' \neq x$ & $h(x') = h(x)$

if finding this is hard then h is second preimage resistant hash fn.

Problem 3 (Collision finding Problem)

$h: P \to S$

find $x, x' \in P$ s.t $x \neq x'$ & $h(x) = h(x')$

- if finding is hard → collision resistant hash fn

__/__/____

Ideal hash fn

Let $h: P \to S$

h is ideal if given $x \in P$ to find $h(x)$ either you have to apply h on x or you have to look into the table corresponding to h (hash table)

Preimage finding algo. given $y \in Y$ find $x \in X$ s.t $h(x) = y$

$h: X \to Y$

choose any $X_0 \subseteq X$ s.t $|X_0| = cq$

for every $x \in X_0$
if $h(x) = y_x$
return x
else chose another subset

so the probability $\propto \frac{1}{\text{complexity}} \propto cq$

E_i: event $h(x_i) = y$: $1 \le i \le cq$

$Pr[E_i] = \frac{1}{m}$

$Pr[E_i'] = 1 - \frac{1}{m}$

$Pr[E_1 \cup E_2 \cup E_3 \dots \cup E_{cq}]$
$= 1 - Pr[E_1' \cap E_2' \cap \dots E_{cq}']$
$= 1 - \left(1 - \frac{1}{m}\right)^{cq} \le 1 - \left[1 - \frac{cq}{m}\right]$
$= \frac{cq}{m}$

$\therefore$ complexity $= O(m)$