Adv. Cryptography

CT-3 CSPE18

106118036

1) BGW protocol.

$p = 11$, $S_1 = 3 + 5x$ $t = 2$

$S_2 = 4 + 2x$

$f(x) = S_1 * S_2 = 12 + 26x + 10x^2$

Now $\underline{\underline{2t - 1 = 3}}$

$f(1) = 4 \% 11 = 4$

$f(2) = 104 \% 11 = 5$

$f(3) = 180 \% 11 = 4$

So,

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{bmatrix}$$

So, $$A^{-1} = \begin{bmatrix} 3 & -5/2 & 1/2 \\ -3 & 4 & -1 \\ 1 & -3/2 & 1/2 \end{bmatrix}$$

$h_1(x) = 8 * 6 + x = 4 + x$

Shares $\rightarrow \{5, 6, 7\}$

$h_2(x) = 13 * 8 + x = 5 + x$

Shares $\rightarrow \{6, 7, 8\}$

$h_3(x) = 18 * 10 + x = 4 + x$

Shares $\rightarrow \{5, 6, 7\}$

$h(x) = \varphi_1 h_1(x) + \varphi_2 h_2(x) + \varphi_3 h_3(x)$

$h(1) = 3 * 5 + 6(-3) + 5$

$= 2$

$h(2) = 3 * 6 + (-3) * 7 + 6$

$= 3$

So, we get $\underline{(1,2)}$ 4 $\underline{(2,3)}$

Using lagrange theorem,

$$Sol^n = \frac{2 * -2}{(1-2)} + \frac{3(-1)}{(2-1)}$$

$$= 4 + (-3)$$

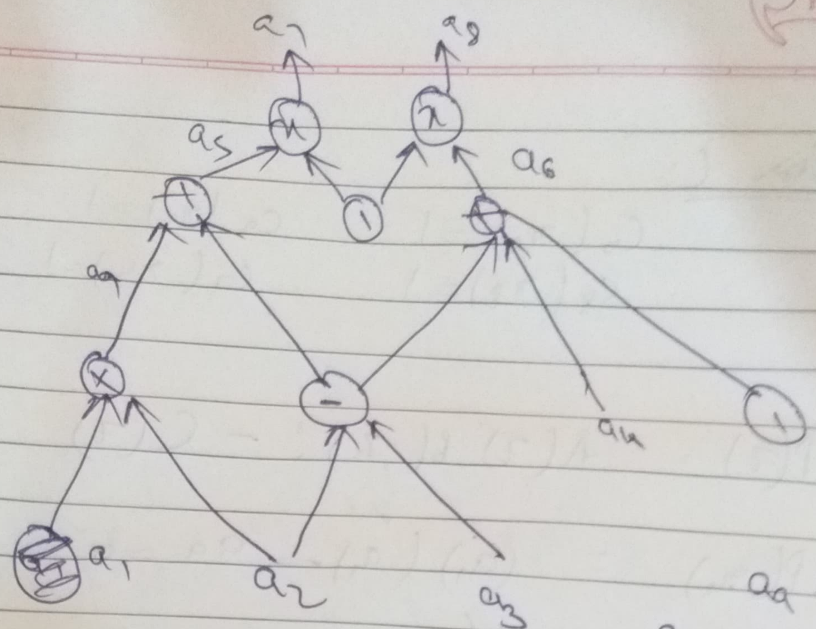
$$= \boxed{1}$$

$S_1 S_2 = 12 \times 11 = \boxed{1}$

$\therefore$ The protocol was demonstrated and verified

Date
Page

2.>



$a_9 = a_1 a_2$

$a_7 = a_5 = a_1 a_2 + a_2 - a_3$

$a_8 = a_6 = a_2 - a_3 + a_4$

3 AND Gates : r_7, r_8, r_9
I/P : $a_1\ a_2\ a_3\ a_4$] assignment

$Z(z) = \deg(3)$
$= (z - r_7)(z - r_8)(z - r_9) \rightarrow$ 1

For A:

$A_0(r_8) = 1$ $\quad A_4(r_7) = 1$
$A_1(r_9) = 1$
$A_2(r_7) = 1$
$A_3(r_9) = -1$

For B:

$B_0(r_7) = 1$
$B_2(r_8) = 1$
$B_3(r_8) = -1$
$B_4(r_8) = 1$

Date
Page

For C:-

$C_0(r_8) = 1$ $C_9(r_2) = 1$

$C_8(r_8) = 1$ $C_7(r_7) = 1$

$$P(z) = A(z)\,B(z) - C(z)$$

$$P(r_a) = (a_1)(a_1) - a_9 = 0$$

$$P(r_7) = (a_1 - a_3 + a_9)(1) - a_7 = 0$$

$$P(r_8) = (1)(a_2 - a_3 + a_4) - (1 + a_8) = 0$$

we use lagrange now to find $P(z)$

$$Z(z) = (z - r_7)(z - r_8)(z - r_9)$$

Date
Page

4.) pragma solidity 0.6.8;

```
import "@/open-zeppelin/contracts/math/SafeMath.sol";
import "../../Interfaces/IItem.sol";
  // Interface has price stock & name

contract OnlineMarket {
    using SafeMath for uint256;
        address owner;
        ~~IItem~~ public item;
        ~~address~~
        struct Item {
            uint id;
            string name;
        }
    // Mapping between item and price
        mapping (Item => uint) priceVals;
    // mapping between item & stock
        mapping (Item => uint) stockVals;
        mapping (address => uint) paidByUser;

function OnlineMarket () {
    owner = msg.sender;
}

function Price (uint NewPrice, uint id, string
        name, uint stockown)
    // set new price ~~for~~ item first check owner
    require (msg.sender == owner);
     priceVals(Item (id, name)) = NewPrice
```

Date
Page

```
    stockvals (Item (id, name)) = stockCount;
}
```

b)

```
function Buy (uint quant, uint id, string name)
{ require (quant * item (id, name) >= msg.value
  paidByUser (msg.sender) += msg.value;
  stockvals (Item (id, name)) = quant;
  owner.send (msg.value);
}
} require (quant <= Item (id, name));
```

~~b) function Buy (uint quant) public~~

c) A possible attack that can take place is that there can be simultaneous ordering/buy due to same time ordering by two people.

- Payable function can be used to exploit and raid the contracts.
- Reentrance vulnerability is also an issue.

Adv. Cryptography

106118036 CT-3 CSPE18

3> $N = 5 * 11 = 55$

$G = \{a : x^2 \equiv a \pmod N\}$

Given that set $(G, * \bmod N)$ is cyclic group.

Now, we need perfect squares, so, it is quadratic residue.

~~$x^2 \in \{1, 4, 9, 16, 25, 36, 49\}$~~

$x^2 \in \{1, 4, 9, 16, 25, 36, 49 \ldots\}$

Therefore, we can have 'a' as,

$a \in \{1, 4, 5, 9, 11, 14, 15, 16, 20, 25, 26, 31, 34, 36, 44, 45, 49\}$

($\because$ Done using code)

Hence

Cyclic Group $G = \{1, 4, 5, 9, 11, 14, 15, 16, 20, 25, 26, 31, 34, 36, 44, 45, 49\}$

But, we are unable to find the generator, hence, No Generator