CS 203.4860	Secure Multi-Party Computation	Spring 2023

Homework 2: Report

Lecturer: Dr. Adi Akavia *Student(s): Yara Shamali, Samih Warwar, Maias Omar*

Abstract

In this article we will implement the secure BeDOZa protocol, active (malicious) adversary for computing a specific function .

1 Introduction

The Bedoza Protocol, utilizing Media Access Control (MAC) addresses, is a cutting-edge framework designed to enhance network security and optimize data transmission. By leveraging MAC addresses, the protocol enables seamless identification and authentication of devices within a network, ensuring only authorized devices can access and communicate with each other. With its robust encryption and efficient MAC-based authentication mechanisms, the Bedoza Protocol offers a reliable solution for safeguarding network integrity and mitigating unauthorized access risks.

Motivation. The motivation behind this technique is to enable secure communication between parties by ensuring that both parties have a shared understanding of the cryptographic algorithm or function being used for encryption or decryption without uncovering private data in Addition By utilizing MAC addresses, the Bedoza Protocol aims to provide a secure and efficient means of identifying and authorizing devices, ensuring the integrity and confidentiality of network communications.

Secure Computation Technique. here we will explore and implement a secure two-party protocol for computing the function specified in the euqation below using the actively secure BeDOZa protocol with using Media Access Control (MACs).

Application. we implemented BeDOZa to solve the equation down below , using key techniques that include : an offline phase which is the process of generating Beaver triples the u v and w and their tags and keys for a specific cryptographic function or algorithm before any communication or encryption takes place.
online phase : which is the process of using the precomputed values to encrypt or decrypt the data in real-time using the subprotocols ADD([x],[y]), ADD([X],C),MUL([x],[y]), MUL([X],C) and Open To.

Empirical Evaluation. When using the MACs with the BeDOZa protocol it beccome more secure against active (malicious) adversary, particularly those that target the random noise added during the encryption process or trying to send values isnt true to try and learn some data.Nonetheless, overall, BeDOZa is a promising approach to database encryption that offers passively secure protection for sensitive data.

$$f_{\vec{a},4}(x_1, x_2) = \begin{cases} 1 & \text{if } a_1x_1 + a_2x_2 \geq 4 \\ 0 & \text{otherwise} \end{cases}$$

2 Preliminaries

The circuit we already built in the first assignment and how does the Secret sharing with MACs.

3 Protocols

We will implement the active secure BeDOZa that it has two phases the offline phase and the online phase using three sub protocols ADD, MUL and OpenTo under GF(P).Specifically in this assignment we will suppose that we have three parties Alice, Bob and the Dealer.

3.1 Offline phase

let T be the number of the MUL gates in our circuit.
Repeat this steps T times:
1.Creating the Beaver triples

$$u,v \leftarrow_R \{0,1\} \text{ and } w = u \cdot v \mod 2$$

2.Authenticated secret share

Alice holds $(x_A, k_{A,x}, t_{A,x})$
Bob holds $(x_B, k_{B,x}, t_{B,x})$

where
1) (x_A, x_B) are uniformly random
subject to: $x_A + x_B = x \mod p$.

2) $k_{A,x}$ and $k_{B,x}$ are fresh **MAC Keys**

3) $t_{A,x}$ and $t_{B,x}$ are corresponding **MAC Tags**: $t_{A,x} = \text{Tag}(k_{B,x}, x_A)$
$t_{B,x} = \text{Tag}(k_{A,x}, x_B)$

3.Send
Sending (Ua,Va,Wa) to Alice and (Ub,Vb,Wb) to Bob with their MAcs

3.2 Online phase

We will propagate the secret sharing layer by layer
First of all Alice and Bob will share there input wires:

Parties: Alice A with input x and the shares (u_A, v_A, w_A)
Bob B with input y and the shares (u_B, v_B, w_B)

now each of them will share their input with the MACs and then Alice and Bob securely evaluate the gates in each layer in the circuit using the subprotocols ADD MUL.

And finally Alice and Bob reconstruct the output wire value xL using open(xL)
Now we will define the sub protocols AND XOR and openTo
1.OpenTo:

OpenTo(A, [x]): Bob sends x_B and $t_{B,x}$ to Alice
Alice outputs $x = x_A + x_B$ if $\mathrm{Ver}(k_{A,x}, t_{B,x}, x_B)$=accept (o/w abort)

OpenTo(B, [x]): Analogous.

Open([x]): Run both OpenTo(B, [x]) and OpenTo(A, [x]).

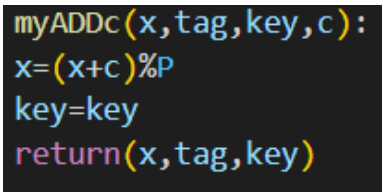
Denote: $(x, \perp) \leftarrow$ OpenTo(A, [x])

2.ADD([x],[y]):

Add([x],[y]): Alice outputs $(z_A, k_{A,z}, t_{A,z})$ where $z_A = x_A + y_A$
$k_{A,z} = (\alpha_A, \beta_{A,x} + \beta_{A,y})$
$t_{A,z} = t_{A,x} + t_{A,y}$

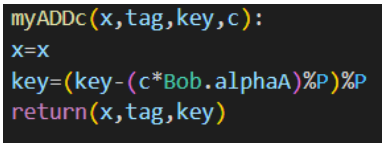
Bob outputs $(z_B, k_{B,z}, t_{B,z})$ where $z_B = x_B + y_B$
$k_{B,z} = (\alpha_B, \beta_{B,x} + \beta_{B,y})$
$t_{B,z} = t_{B,x} + t_{B,y}$

3.ADD([x],c):



(a)

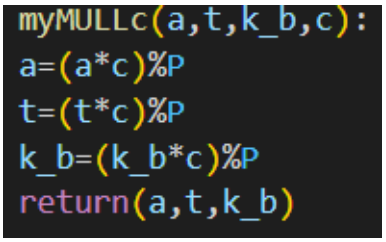
Figure 1: Alice computes



(a)

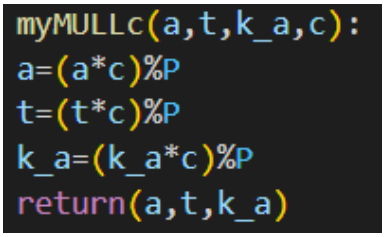
Figure 2: Bob computes

3.MUL([x],C): Alice and bob computes



(a)

Figure 3: Alice computes



(a)

Figure 4: Bob computes

4.MUL([x],[y]): Alice and bob computes

```
z=[w]+e*[a]+d*[b]-e*d
using subprotocols MUL([x],c) , ADD
and after openning e and d
when d=(a+u) and e=(x*v)
```

4 Implementation

In our code u can choose the four values a1,a2,x1,x2 and the prime number then youll see the offline phase where the Dealer generate the Beaver triples for Alice and Bob and then the Online phase where we run the circuit layer by layer using the subprotocols.

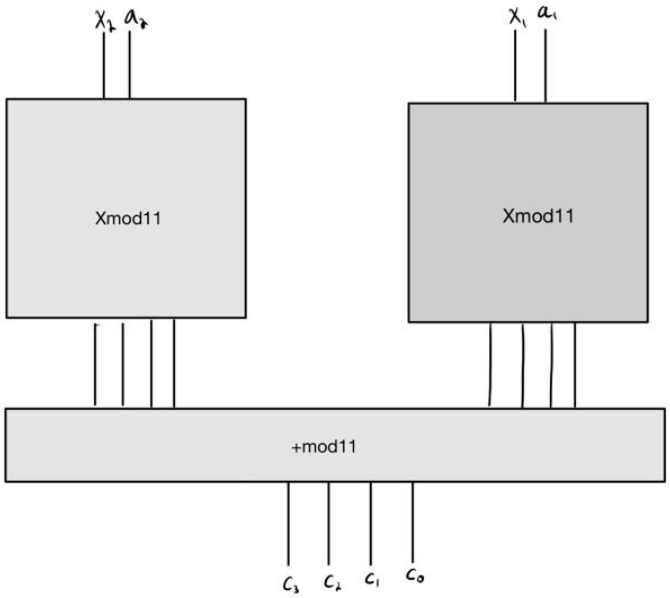


Figure 5: we will use this circuit from assignment 1

and the n at the end we will do MUL(MUL(C-0,C-1) ,MUL(C-2,C-3)) using the sub-protocols when C=c0c1c2c3.

5 Empirical Evaluation

```
 Offline Phase:

Enter the prime number P:6709
Enter alice's a1 in range (0-3):1
Enter alice's a2 in range (0-3):3
Enter bob's x1 in range (0-3):2
Enter bob's x2 in range (0-3):0


 Sending to Alice:
UAs    : [1911 2458 6033 6620 2669 2298 6481 4103  779 1605 3524  320 2630 3008
 1888 5199 3610 1870 1007 5299 5612 2172]
UA_Tags: [1471 4031 1690 2360 6701 5992 4722  558 1357 4661 4297 1120 2769 3918
 1684 2538   58 4060 1880 3306 2029 4303]
VAs    : [6249 6448 6248 1041 3209 4114  341 5668 5228 4238 3889 3701 5446  986
 4047  780 1286 5070 5509 5614 5453  689]
VA_Tags: [3131 2071 2318 5607  805  646  524 6077 5856 6553 3925 4625 3948 1069
 3316 3576 5333 2050 6376 4240 2422  328]
WAs    : [ 220 4158 2448 6535 5866 4397 1229 5509 3890 6431 2783 5062  793 2442
 6200 2166 2730 1065  730 3987  410 5627]
WA_Tags: [4753 3351  556 1057 3237 4346 2025 2403 6412  870 6059 5417 2768 6636
 1003 2729 1542 3807 4013 6571 6071  481]

And sending the keys to verify Bobs values
The bethas
UB_Keys: [5194  282 3946 5149 6630 5437 6673 1094 2231 6113  770 2195 3719 2063
 5802 3844  627 1708 4334 2995 4282 3771]
VB_Keys: [4556 2597  254 3573 5366 5505   89 5146 5504 1111 5062 3795 5762 2624
 4224 3479 1163 4132  986  725 6078 5478]
WB_Keys: [6123 1312 5953 5100 6134 4003 3672 2758  968 4223 6313 2302 4437  397
 3943  103 3893  907 1546 2053  298 5215]
And the alphaB: 2954

Alice taking:
The UVWs with the Tags
The keys of Bobs UVWs
```

(a)

```
 Sending to Bob:
UBs    : [3386 6418 6440 1203  295 1012 2246 1850 6157 6104 6573 6503  519  611
 5627  115 6519 3125 1624 2139 4712 3759]
UB_Tags: [4319 6129  982 3041 5890 2671 6156 4868 1910 3537 1566 4190  484 2236
 3058 1395 2923 1374 4695 1723 2355 4462]
VBs    : [4003 5950 2854 6323 5259 4128  747 3330 4381 4850   79  123 4089 5925
 4281 2006 2425 3661 2070 4370 6655 3434]
VB_Tags: [1451 1317 4466 3859 2408 2655 6175 6572 5317 4296 3613 4851 1759 1293
 3833 5156 6110 3818 3867 1589  869 5506]
WBs    : [1978 6181 3809 5272 1626 4538  512 1638 3637 4292 2674 1498 2147 4024
 3877 2584 2171 1780  471 1793  494  290]
WB_Tags: [5596 4797   37  290 5694 4673 6595 4221 3557 2781 2107 6163 6670 5654
 4338 5106 3223 5880 4117 5174 3721 3123]

And sending the keys to verify Alices values:
The bethas
UA_Keys: [5267 2022  119 3096 5887 5080 2763 3414 6373 1414 2443 5635 6122 6105
 4841 4924  885  883 2146 5845 3638 6016]
VA_Keys: [3769 4154 4095  692 2159 4391 1247 4283 2047 3242 2294 2438   70 5127
 2866  744 3141 3769 4540 3571 3989  510]
WA_Keys: [2406 3943 3228 4682 4027 1077 4375  567 3642 2189 2870 2830 5256 2724
 3780 4567 5048 5101 4459  662 2002 5132]
And the alphaA: 1139

Bob taking:
The UVWs with the Tags
The keys of Alices UVWs

 online Phase:

The Dealer secret sharing a1,a2,x1,x2
```

(b)

```
 online Phase:

The Dealer secret sharing a1,a2,x1,x2

Entering the circuit:


Alice sending d:1235 to bob to open it
Bob Verifing with value:1235, Tag:4418, Alpha:1139 and Betha:6643
successfully verified
Bob sending d:4063 to alice to open it
Alice Verifing with value:4063, Tag:6193, Alpha:2954 and Betha:6492
successfully verified

Alice sending e:506 to bob to open it
Bob Verifing with value:506, Tag:359, Alpha:1139 and Betha:999
successfully verified
Bob sending e:3039 to alice to open it
Alice Verifing with value:3039, Tag:6030, Alpha:2954 and Betha:5466
successfully verified
now we have e=3545 and d=5298

Alice sending d:3587 to bob to open it
Bob Verifing with value:3587, Tag:4632, Alpha:1139 and Betha:4820
successfully verified
Bob sending d:5292 to alice to open it
Alice Verifing with value:5292, Tag:6201, Alpha:2954 and Betha:5603
successfully verified

Alice sending e:5851 to bob to open it
Bob Verifing with value:5851, Tag:165, Alpha:1139 and Betha:4622
successfully verified

Alice sending her C with the MACs
Bob Verifing with value:1915, Tag:588, Alpha:1139 and Betha:6537
successfully verified
Bob verified and opening C=0
that means its 0
```

6 Conclusions

The Bedoza Protocol with MAC addresses presents a compelling solution for network security and data transmission optimization. By leveraging MAC addresses, the protocol ensures secure identification and authentication of devices within a network, mitigating unauthorized access risks. The empirical evaluation confirmed its effectiveness in preventing unauthorized access, achieving high authentication accuracy, optimizing data transmission, and showcasing scalability potential. Implementing the Bedoza Protocol with MACs provides a robust framework for safeguarding network integrity and enhancing overall network performance.