CYBERNETICA

The design and implementation of a two-party protocol suite for Sharemind

Pille Pullonen, Dan Bogdanov, Thomas Schneider

Motivation

- Sharemind 3
  - Secure multi-party computation
- Protection Domains
  - First new domain kind

Overview

- Domain setup
- Necessary protocols
- Multiplication and problems
- Performance
- Remaining issues and future plans

General setup

- Semi-honest
- Additive secret sharing in $Z_{2^{32}}$
- Two parties P1 and P2
- Input and output parties
- Encrypted and authenticated channels
- Additively homomorphic cryptosystem
- P1 has a keypair (pk,sk), P2 knows pk

Additively homomorphic cryptosystem

- $E_{pk}(x)$, $D_{sk}(E_{pk}(x)) = x$
- $E_{pk}(x+y) = E_{pk}(x)E_{pk}(y)$
- $E_{pk}(kx)=E_{pk}(x)^k$, k is public
- Given $E_{pk}(x)$, $E_{pk}(y)$: $E_{pk}(xy)=?$
- Paillier cryptosystem
- 2048-bit key, 4096-bit ciphertexts

Additive secret sharing

- Value a
  - Shares a1, a2
  - a = a1 + a2
- Computation
  - Addition, multiplication
- Supporting protocols
  - Sharing, publishing, resharing

Resharing a to b

- P1
  - Has a1
  - Generates r1 $\leftarrow Z_{2^{32}}$
  - Receive r2
  - b1 = a1 + r1 - r2
- P2
  - Has a2
  - Generates r2 $\leftarrow Z_{2^{32}}$
  - Receive r1
  - b2 = a2 + r2 - r1

$$a = a1 + a2 =$$
$$a1 + a2 + r1 - r1 + r2 - r2 =$$
$$b1 + b2 = b$$

Classify

- Input party
  - Secret a
  - Generate a2 ← $Z_{2^{32}}$
  - a1 = a - a2

- P1
  - Receive share a1

- P2
  - Receive share a2

Declassify

- P1
  - Share a1
  - Share t1

Reshare a to t

- P2
  - Share a2
  - Share t2

- Output parties

output = t1 + t2 = a

Declassify

- P1
  - Share a1
  - Share t1

Reshare a to t?

- P2
  - Share a2
  - Share t2

- Output parties
  $a = t1 + t2$

- Parties, except P1 and P2, don't learn a1, a2

Addition and subtraction

- P1
  - Shares a1, b1
  - Add:
    - $c1 = a1 + b1$
  - Subtract:
    - $c1 = a1 - b1$
- P2
  - Shares a2, b2
  - Add:
    - $c2 = a2 + b2$
  - Subtract:
    - $c2 = a2 - b2$

$$a = a1 + a2,\ b = b1 + b2$$
$$a + b = a1 + a2 + b1 + b2$$
$$a - b = a1 + a2 - b1 - b2$$

Multiplication

- $a = a1 + a2, b = b1 + b2$
- $a \cdot b = (a1+a2) \cdot (b1+b2) =$
  $a1 \cdot b1 + a1 \cdot b2 + a2 \cdot b1 + a2 \cdot b2$

Multiplication

- $a = a1 + a2$, $b = b1 + b2$
- $a \cdot b = (a1+a2) \cdot (b1+b2) =$
  $a1 \cdot b1 + a1 \cdot b2 + a2 \cdot b1 + a2 \cdot b2$
- Additively homomorphic cryptosystem
  - $E(a1)^{b2} = E(a1 \cdot b2)$
  - $D(E(a1 \cdot b2)) = a1 \cdot b2$

Multiplication

- P1
  - Keypair (pk,sk)
  - Shares a1, b1
  - Compute E(a1), E(b1) →
  - ← t = D(E(t))
  - c1 = a1b1 + t mod 2^{32}
- P2
  - Key pk
  - Shares a2, b2
  - E(a1), E(b1)
  - Generate r $\leftarrow \{0,1\}^{2\cdot 32+1+k}$
  - E(r), E(a1•b2), E(a2•b1)
  - E(t)=(a1•b2+a2•b1+r)
  - c2 = a2b2 – r mod 2^{32}

c = c1 + c2 = a1b1 + a1b2 + b2b1 + r + a2b2 – r = ab

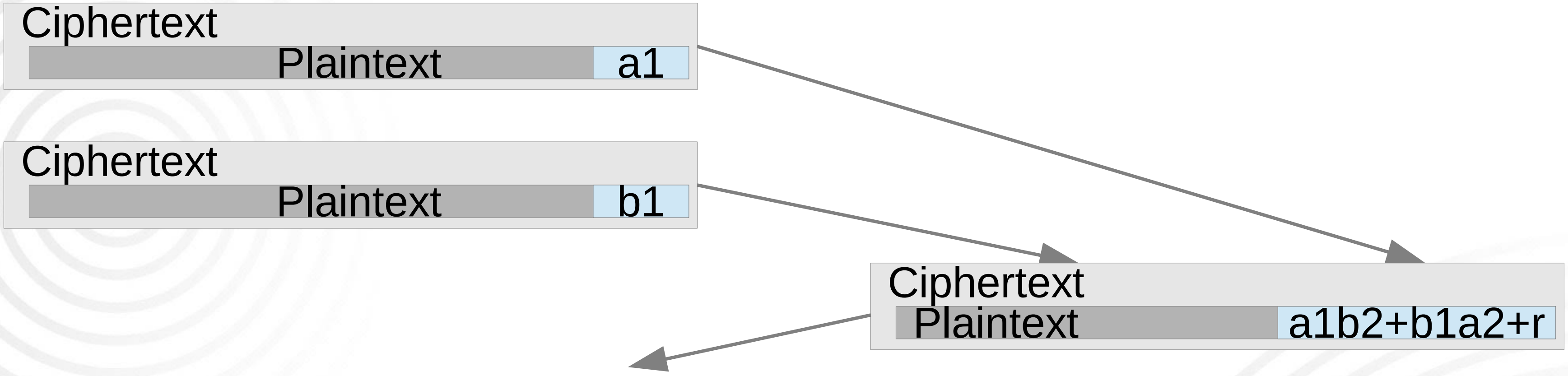
Problems with multiplication

- 32-bit values, 4096-bit ciphertext
- Possible 2048-bit plaintext

- Inefficient communication
- Vectors

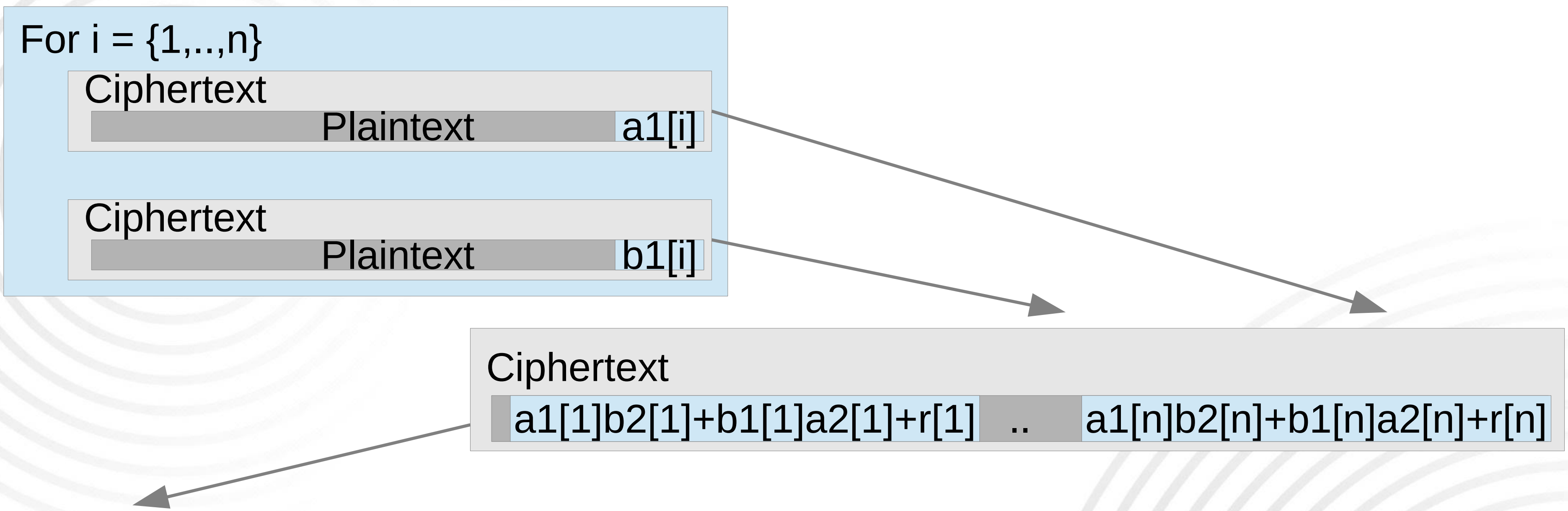
Multiplication

- P1
- P2



Packing vectors into one ciphertext

- P1
- P2



2048-bit key, 32-bit values, and k = 112 give n = 11

CYBERNETICA

Multiplication on vectors

- P1
  - Shares a1[], b1[]
  - E(a1[1]),..,E(a1[n]) →
  - E(b1[1]),..,E(b1[n]) →
  - t[1] || ..|| t[n] = D(E(t)) ←
  - c1[i]=a1[i]b1[i] + t[i] mod 2^{32}
- P2
  - Shares a2[], b2[]
  - t[i] = a1[i]b2[i] + a2[i]b1[i] + r[i]
  - L = the max length of t[i]
  - t = t[1]*$2^{(n-1)L}$ + .. + t[n]
  - E(t)=E(t[1] || ..|| t[n])
  - c2[i] = a2[i]b2[i] – r[i] mod 2^{32}

Remaining problems

- Encryption is slow
  - P1 must encrypt all inputs
- This protocol is slow
  - Don't want to wait during computation

Precomputation and online multiplication

- This protocol gives c = ab
  - Stored as shares
- Can precompute random triples
- Fast online phase using the triples
- Long startup
  - Feasible for some settings

Online multiplication

- P1
  - Shares $a1, b1$
  - $w1, x1, y1$
  - $c1 = w1 + ey1 + zx1 + ez$

Choose a triple $w = xy$

Subtract
$e = a-x$
$z = b-y$

Declassify e, z

- P2
  - Shares $a2, b2$
  - $w2, x2, y2$
  - $c2 = w2 + ey2 + zx2$

$$c = c1 + c2 = w + (a-x)y + (b-y)x + (a-x)(b-y) =$$
$$= xy + ay - xy + bx - yx + ab - ay - xb + xy = ab$$

Performance

	Three parties		Two parties			
length	addition	multiply	addition	triples	multiply P1	Multiply P2
1000	0.033	27.538	0.03	37402.9	21.061	0.201
10000	0.309	59.717	0.298	376045.8	25.239	5.277
20000	0.603	91.396	0.571	754546.6	30.649	13.34
100000	2.556	414.367	2.548	3759854.9	94.252	81.104

(milliseconds)

CYBERNETICA

Future work

- More efficient packing
- Symmetric multiplication precomputation
- Faster encryption
- More protocols
- Active security model

References

- research.cyber.ee
  The design and implementation of a two-party protocol suite for Sharemind
- http://sharemind.cyber.ee/