How to Implement Anything in MPC

Marcel Keller Peter Scholl Nigel Smart

University of Bristol

19 February 2015

Overview

1. Compiler for MPC circuits
2. Implement ORAM as MPC circuit
   $\Rightarrow$ Oblivious array / memory
3. Execute a set of instructions (small circuits) at every step
   $\Rightarrow$ Oblivious execution

Oblivious machine

- 40 Hz with 1000-entry memory
- 2 Hz with 10^6-entry memory

Problem:

Overview

1. Compiler for MPC circuits
2. Implement ORAM as MPC circuit
   $\Rightarrow$ Oblivious array / memory
3. Execute a set of instructions (small circuits) at every step
   $\Rightarrow$ Oblivious execution

Oblivious machine

- 40 Hz with 1000-entry memory
- 2 Hz with 10^6-entry memory

Problem: "The idea of implementing ORAM within the MPC is not new, and figuring out the details while cumbersome, doesn't seem to require any new ideas or techniques. I was not able to pin down any such ideas or any surprises when implementing ORAM in MPC."

Overview

1. Compiler for MPC circuits
2. Implement ORAM as MPC circuit
   ⇒ Oblivious array / memory
3. Execute a set of instructions (small circuits) at every step
   ⇒ Oblivious execution

Oblivious machine

- 40 Hz with 1000-entry memory
- 2 Hz with 10^6-entry memory

Problem: "The idea of implementing ORAM within the MPC is not new, and figuring out the details while cumbersome, doesn't seem to require any new ideas or techniques. I was not able to pin down any such ideas or any surprises when implementing ORAM in MPC."

~~How to Implement Anything in MPC~~
Malicious-for-free OT Extension and Beyond

All You Can OT

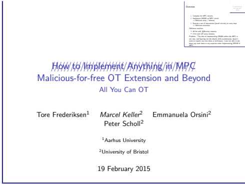
Tore Frederiksen[1] *Marcel Keller*[2] Emmanuela Orsini[2]
Peter Scholl[2]

[1]Aarhus University

[2]University of Bristol

19 February 2015

Overview


~~How to Implement Anything in MPC~~
Malicious-for-free OT Extension and Beyond
All You Can OT

Tore Frederiksen[1] *Marcel Keller*[2] Emmanuela Orsini[2]
Peter Scholl[2]

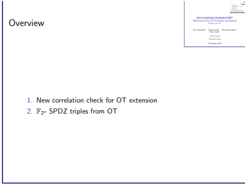
[1]Aarhus University
[2]University of Bristol

19 February 2015

1. New correlation check for OT extension
2. $\mathbb{F}_{2^n}$ SPDZ triples from OT

OT Extension

1. κ base OTs
2. Extend length with pseudorandom functions
3. Introduce correlation
4. Transpose
5. Hash to break correlation



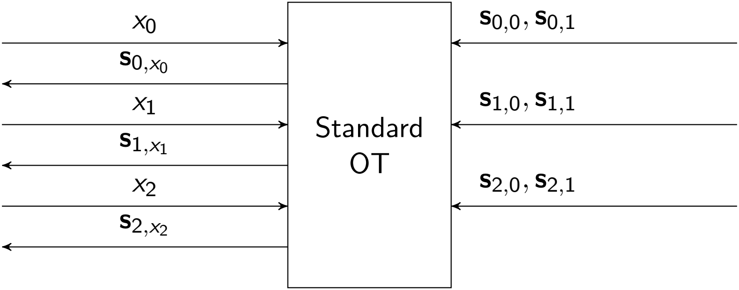
OT Extension

1. κ base OTs
2. Extend length with pseudorandom functions
3. Introduce correlation
4. Transpose
5. Hash to break correlation

Overview

1. New correlation check for OT extension
2. $\mathbb{F}_{2^k}$ SPDZ triples from OT

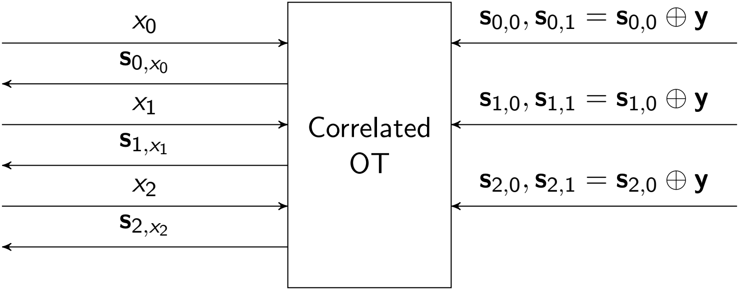
Another Look at Correlated OT



x_i: selection bit

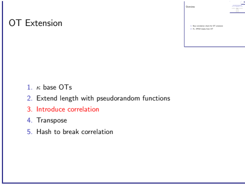
$\mathbf{s}_{i,0}, \mathbf{s}_{i,1}$: strings

Another Look at Correlated OT

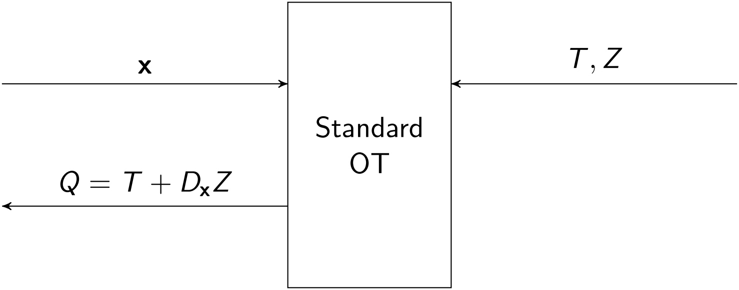


x_i: selection bit

$\mathbf{y}, \mathbf{s}_{i,0}, \mathbf{s}_{i,1}$: strings

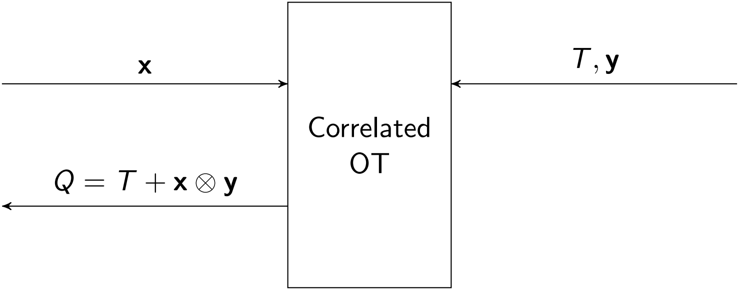


Another Look at Correlated OT



$\mathbf{x}$: string / vector in $(\mathbb{F}_2)^\kappa$
Q, T: matrices in $(\mathbb{F}_2)^{\kappa \times \ell}$
$D_{\mathbf{x}}$: matrix with diagonal $\mathbf{x}$

Another Look at Correlated OT



$\mathbf{x}, \mathbf{y}$: strings / vectors in $(\mathbb{F}_2)^\kappa$

Q, T: matrices in $(\mathbb{F}_2)^{\kappa \times \ell}$

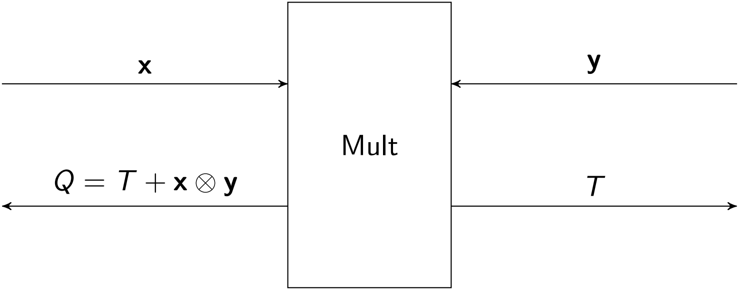
$D_\mathbf{x}$: matrix with diagonal $\mathbf{x}$

$\mathbf{x} \otimes \mathbf{y}$: tensor product, matrix of all possible products

OT Extension

1. κ base OTs
2. Extend length with pseudorandom functions
3. Introduce correlation
4. Transpose
5. Hash to break correlation

Another Look at Correlated OT

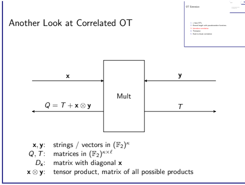


$\mathbf{x}, \mathbf{y}$: strings / vectors in $(\mathbb{F}_2)^\kappa$

Q, T: matrices in $(\mathbb{F}_2)^{\kappa \times \ell}$

$D_{\mathbf{x}}$: matrix with diagonal $\mathbf{x}$

$\mathbf{x} \otimes \mathbf{y}$: tensor product, matrix of all possible products



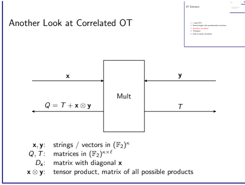
Errors in Correlation

Honest receiver:

$$Q + T = D_{\mathbf{x}} Z = \mathbf{x} \otimes \mathbf{y} =$$

$$\begin{pmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 & x_1 y_4 & x_1 y_5 & x_1 y_6 & x_1 y_7 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 & x_2 y_4 & x_2 y_5 & x_2 y_6 & x_2 y_7 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 & x_3 y_4 & x_3 y_5 & x_3 y_6 & x_3 y_7 \\ x_4 y_1 & x_4 y_2 & x_4 y_3 & x_4 y_4 & x_4 y_5 & x_4 y_6 & x_4 y_7 \\ x_5 y_1 & x_5 y_2 & x_5 y_3 & x_5 y_4 & x_5 y_5 & x_5 y_6 & x_5 y_7 \\ x_6 y_1 & x_6 y_2 & x_6 y_3 & x_6 y_4 & x_6 y_5 & x_6 y_6 & x_6 y_7 \\ x_7 y_1 & x_7 y_2 & x_7 y_3 & x_7 y_4 & x_7 y_5 & x_7 y_6 & x_7 y_7 \end{pmatrix}$$

Errors in Correlation



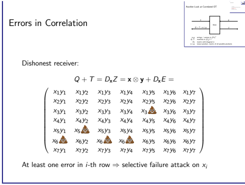
Dishonest receiver:

$$Q + T = D_{\mathbf{x}} Z = \mathbf{x} \otimes \mathbf{y} + D_{\mathbf{x}} E =$$

$$\begin{pmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 & x_1 y_4 & x_1 y_5 & x_1 y_6 & x_1 y_7 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 & x_2 y_4 & x_2 y_5 & x_2 y_6 & x_2 y_7 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 & x_3 y_4 & x_3 \text{💩} & x_3 y_6 & x_3 y_7 \\ x_4 y_1 & x_4 y_2 & x_4 y_3 & x_4 y_4 & x_4 y_5 & x_4 y_6 & x_4 y_7 \\ x_5 y_1 & x_5 \text{💩} & x_5 y_3 & x_5 y_4 & x_5 y_5 & x_5 y_6 & x_5 y_7 \\ x_6 \text{💩} & x_6 y_2 & x_6 \text{💩} & x_6 \text{💩} & x_6 y_5 & x_6 y_6 & x_6 y_7 \\ x_7 y_1 & x_7 y_2 & x_7 y_3 & x_7 y_4 & x_7 y_5 & x_7 y_6 & x_7 y_7 \end{pmatrix}$$

At least one error in i-th row $\Rightarrow$ selective failure attack on x_i

Correlation Check



Honest receiver: $Q + T = \mathbf{x} \otimes \mathbf{y}$

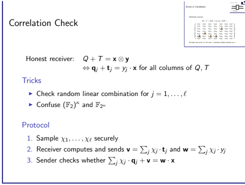
$\Leftrightarrow \mathbf{q}_j + \mathbf{t}_j = y_j \cdot \mathbf{x}$ for all columns of Q, T

Tricks

- Check random linear combination for $j = 1, \ldots, \ell$
- Confuse $(\mathbb{F}_2)^\kappa$ and $\mathbb{F}_{2^\kappa}$

Protocol

1. Sample $\chi_1, \ldots, \chi_\ell$ securely
2. Receiver computes and sends $\mathbf{v} = \sum_j \chi_j \cdot \mathbf{t}_j$ and $\mathbf{w} = \sum_j \chi_j \cdot y_j$
3. Sender checks whether $\sum_j \chi_j \cdot \mathbf{q}_j + \mathbf{v} = \mathbf{w} \cdot \mathbf{x}$



Correlation Check

Honest receiver: $Q + T = \mathbf{x} \otimes \mathbf{y}$
$\Leftrightarrow \mathbf{q}_j + \mathbf{t}_j = y_j \cdot \mathbf{x}$ for all columns of Q, T

Tricks

- Check random linear combination for $j = 1, \ldots, \ell$
- Confuse $(\mathbb{F}_2)^\kappa$ and $\mathbb{F}_{2^\kappa}$

Protocol

1. Sample $\chi_1, \ldots, \chi_\ell$ securely
2. Receiver computes and sends $\mathbf{v} = \sum_j \chi_j \cdot \mathbf{t}_j$ and $\mathbf{w} = \sum_j \chi_j \cdot y_j$
3. Sender checks whether $\sum_j \chi_j \cdot \mathbf{q}_j + \mathbf{v} = \mathbf{w} \cdot \mathbf{x}$

Correlation Check II

Dishonest receiver: $Q + T = D_{\mathbf{x}} Z$

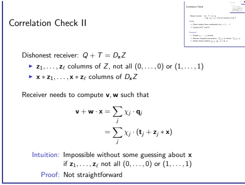
- $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ columns of Z, not all $(0, \ldots, 0)$ or $(1, \ldots, 1)$
- $\mathbf{x} * \mathbf{z}_1, \ldots, \mathbf{x} * \mathbf{z}_\ell$ columns of $D_{\mathbf{x}} Z$

Receiver needs to compute $\mathbf{v}, \mathbf{w}$ such that

$$\begin{aligned} \mathbf{v} + \mathbf{w} \cdot \mathbf{x} &= \sum_j \chi_j \cdot \mathbf{q}_j \\ &= \sum_j \chi_j \cdot (\mathbf{t}_j + \mathbf{z}_j * \mathbf{x}) \end{aligned}$$

Intuition: Impossible without some guessing about $\mathbf{x}$
if $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ not all $(0, \ldots, 0)$ or $(1, \ldots, 1)$

Proof: Not straightforward


Correlation Check II

Dishonest receiver: $Q + T = D_{\mathbf{x}} Z$
- $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ columns of Z, not all $(0, \ldots, 0)$ or $(1, \ldots, 1)$
- $\mathbf{x} * \mathbf{z}_1, \ldots, \mathbf{x} * \mathbf{z}_\ell$ columns of $D_{\mathbf{x}} Z$

Receiver needs to compute $\mathbf{v}, \mathbf{w}$ such that

$$\begin{aligned}\mathbf{v} + \mathbf{w} \cdot \mathbf{x} &= \sum_j \chi_j \cdot \mathbf{q}_j \\ &= \sum_j \chi_j \cdot (\mathbf{t}_j + \mathbf{z}_j * \mathbf{x})\end{aligned}$$

Intuition: Impossible without some guessing about $\mathbf{x}$ if $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ not all $(0, \ldots, 0)$ or $(1, \ldots, 1)$

Proof: Not straightforward

Correlation Check III

Receiver needs to compute w such that

$$\mathbf{v} + \mathbf{w} \cdot \mathbf{x} = \sum_j \chi_j \cdot (\mathbf{t}_j + \mathbf{z}_j * \mathbf{x})$$

Example

$\mathbf{z}_j = (1, 0, 1, 0, \ldots, 1, 0)$ for all j. If $x_1 = x_2, x_3 = x_4, \ldots,$

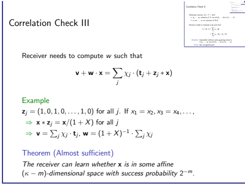
$\Rightarrow$ $\mathbf{x} * \mathbf{z}_j = \mathbf{x}/(1 + X)$ for all j

$\Rightarrow$ $\mathbf{v} = \sum_j \chi_j \cdot \mathbf{t}_j$, $\mathbf{w} = (1 + X)^{-1} \cdot \sum_j \chi_j$

Theorem (Almost sufficient)

The receiver can learn whether $\mathbf{x}$ *is in some affine* $(\kappa - m)$*-dimensional space with success probability* 2^{-m}*.*

Correlation Check – The Other Side


Correlation Check III

Receiver needs to compute w such that

$$\mathbf{v} + \mathbf{w} \cdot \mathbf{x} = \sum_j \chi_j \cdot (\mathbf{t}_j + \mathbf{z}_j * \mathbf{x})$$

Example

$\mathbf{z}_j = (1, 0, 1, 0, \ldots, 1, 0)$ for all j. If $x_1 = x_2, x_3 = x_4, \ldots,$

$\Rightarrow$ $\mathbf{x} * \mathbf{z}_j = \mathbf{x}/(1 + X)$ for all j

$\Rightarrow$ $\mathbf{v} = \sum_j \chi_j \cdot \mathbf{t}_j$, $\mathbf{w} = (1 + X)^{-1} \cdot \sum_j \chi_j$

Theorem (Almost sufficient)

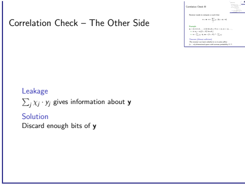
The receiver can learn whether $\mathbf{x}$ *is in some affine* $(\kappa - m)$*-dimensional space with success probability* 2^{-m}.

Leakage

$\sum_j \chi_j \cdot y_j$ gives information about $\mathbf{y}$

Solution

Discard enough bits of $\mathbf{y}$



Experiments

- 10 million OTs
- 8 threads

	LAN	WAN
Passive security	3.3258 s	13.1510 s
Active security	3.3516 s	13.4157 s

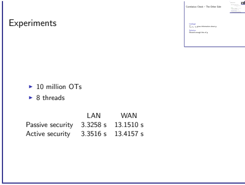
Experiments

- 10 million OTs
- 8 threads

	LAN	WAN
Passive security	3.3258 s	13.1510 s
Active security	3.3516 s	13.4157 s

Part II

Beyond – The Road to SPDZ


Experiments

- 10 million OTs
- 8 threads

	LAN	WAN
Passive security	3.3258 s	13.1510 s
Active security	3.3516 s	13.4157 s

Amplified Correlated OT

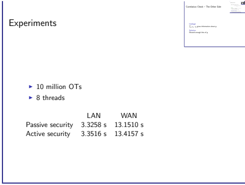
TinyOT: Check correlated OT, then amplify

SPDZ: Check in sacrificing, amplify first?

Amplification with random matrix $M \in \mathbb{F}_2^{\ell \times 3\ell}$ (honest receiver):

$$\begin{aligned} MQ + MT &= MD_{\mathbf{x}}Z \\ &= M \cdot (\mathbf{x} \otimes \mathbf{y}) \\ &= (M\mathbf{x}) \otimes \mathbf{y} \end{aligned}$$

Amplified Correlated OT


Experiments

- 10 million OTs
- 8 threads

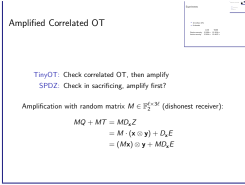
	LAN	WAN
Passive security	3.3258 s	13.1510 s
Active security	3.3516 s	13.4157 s

TinyOT: Check correlated OT, then amplify

SPDZ: Check in sacrificing, amplify first?

Amplification with random matrix $M \in \mathbb{F}_2^{\ell \times 3\ell}$ (dishonest receiver):

$$\begin{aligned} MQ + MT &= MD_{\mathbf{x}}Z \\ &= M \cdot (\mathbf{x} \otimes \mathbf{y}) + D_{\mathbf{x}}E \\ &= (M\mathbf{x}) \otimes \mathbf{y} + MD_{\mathbf{x}}E \end{aligned}$$



Amplification of Errors

Amplification with random matrix $M \in \mathbb{F}_2^{\ell \times 3\ell}$:

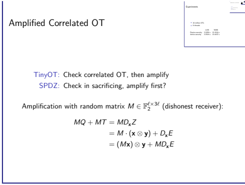
$$MD_{\mathbf{x}}Z = (M\mathbf{x}) \otimes \mathbf{y} + MD_{\mathbf{x}}E =$$

$$M \cdot \begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 & x_1y_4 & x_1y_5 & x_1y_6 & x_1y_7 \\ x_2y_1 & x_2y_2 & x_2y_3 & x_2y_4 & x_2y_5 & x_2y_6 & x_2y_7 \\ x_3y_1 & x_3y_2 & x_3y_3 & x_3y_4 & x_3\text{💩} & x_3y_6 & x_3y_7 \\ x_4y_1 & x_4y_2 & x_4y_3 & x_4y_4 & x_4y_5 & x_4y_6 & x_4y_7 \\ x_5y_1 & x_5y_2 & x_5y_3 & x_5y_4 & x_5y_5 & x_5y_6 & x_5y_7 \\ x_6y_1 & x_6y_2 & x_6y_3 & x_6y_4 & x_6y_5 & x_6y_6 & x_6y_7 \\ x_7y_1 & x_7y_2 & x_7y_3 & x_7y_4 & x_7y_5 & x_7y_6 & x_7y_7 \end{pmatrix}$$

Few errors: $M\mathbf{x}$ independent of errors with high probability

Many errors: $MD_{\mathbf{x}}E$ has high entropy

$\Rightarrow$ SPDZ sacrifice will fail with high probability



Amplification of Errors

Amplification with random matrix $M \in \mathbb{F}_2^{\ell \times 3\ell}$:

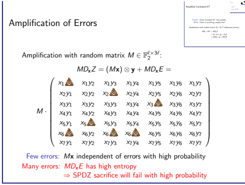
$$MD_{\mathbf{x}}Z = (M\mathbf{x}) \otimes \mathbf{y} + MD_{\mathbf{x}}E =$$

$$M \cdot \begin{pmatrix}
x_1 \text{💩} & x_1 y_2 & x_1 y_3 & x_1 y_4 & x_1 y_5 & x_1 y_6 & x_1 y_7 \\
x_2 y_1 & x_2 y_2 & x_2 \text{💩} & x_2 y_4 & x_2 y_5 & x_2 y_6 & x_2 y_7 \\
x_3 y_1 & x_3 y_2 & x_3 y_3 & x_3 y_4 & x_3 \text{💩} & x_3 y_6 & x_3 y_7 \\
x_4 y_1 & x_4 y_2 & x_4 y_3 & x_4 y_4 & x_4 y_5 & x_4 y_6 & x_4 y_7 \\
x_5 y_1 & x_5 \text{💩} & x_5 y_3 & x_5 y_4 & x_5 y_5 & x_5 y_6 & x_5 y_7 \\
x_6 \text{💩} & x_6 y_2 & x_6 \text{💩} & x_6 \text{💩} & x_6 y_5 & x_6 y_6 & x_6 y_7 \\
x_7 y_1 & x_7 y_2 & x_7 y_3 & x_7 y_4 & x_7 y_5 & x_7 y_6 & x_7 y_7
\end{pmatrix}$$

Few errors: $M\mathbf{x}$ independent of errors with high probability

Many errors: $MD_{\mathbf{x}}E$ has high entropy

$\Rightarrow$ SPDZ sacrifice will fail with high probability



Generating $\mathbb{F}_{2^n}$ SPDZ Triples

Amplified correlated OT is a two-party tensor product box
$\Rightarrow$ Use for every pair of parties

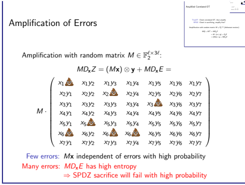
Protocol

1. Pairwise amplified OT to get secret-shared triple $[a], [b], [c = a \cdot b]$
2. Pairwise leaky correlated OT to get secret-shared MACs $[a \cdot \Delta], [b \cdot \Delta], [c \cdot \Delta]$
3. Improved SPDZ sacrifice (error detection)

Running Time

Estimate: 200 times faster than SPDZ for $\mathbb{F}_{2^n}$

Generating $\mathbb{F}_{2^n}$ SPDZ Triples


Amplification of Errors

Amplification with random matrix $M \in \mathbb{F}_2^{\ell \times 3\ell}$:

$$MD_\mathbf{x}Z = (M\mathbf{x}) \otimes \mathbf{y} + MD_\mathbf{x}E =$$

$$M \cdot \begin{pmatrix} x_1\text{💩} & x_1y_2 & x_1y_3 & x_1y_4 & x_1y_5 & x_1y_6 & x_1y_7 \\ x_2y_1 & x_2y_2 & x_2\text{💩} & x_2y_4 & x_2y_5 & x_2y_6 & x_2y_7 \\ x_3y_1 & x_3y_2 & x_3y_3 & x_3y_4 & x_3\text{💩} & x_3y_6 & x_3y_7 \\ x_4y_1 & x_4y_2 & x_4y_3 & x_4y_4 & x_4y_5 & x_4y_6 & x_4y_7 \\ x_5y_1 & x_5\text{💩} & x_5y_3 & x_5y_4 & x_5y_5 & x_5y_6 & x_5y_7 \\ x_6\text{💩} & x_6y_2 & x_6\text{💩} & x_6\text{💩} & x_6y_5 & x_6y_6 & x_6y_7 \\ x_7y_1 & x_7y_2 & x_7y_3 & x_7y_4 & x_7y_5 & x_7y_6 & x_7y_7 \end{pmatrix}$$

Few errors: $M\mathbf{x}$ independent of errors with high probability

Many errors: $MD_\mathbf{x}E$ has high entropy

$\Rightarrow$ SPDZ sacrifice will fail with high probability

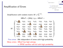
Protocol

1. Pairwise amplified OT to get secret-shared triple
2. Pairwise leaky correlated OT to get secret-shared MACs
3. Improved SPDZ sacrifice (error detection)

Attacks

- Cheating in amplified OT:
  Erased by amplification or fail error detection
- Cheating in leaky correlated OT:
  Some bits of MAC key are leaked. Not an issue because complete leakage succeeds only with negligible probability
- Use different inputs for different parties: fail error detectin

Generating $\mathbb{F}_{2^n}$ SPDZ Triples



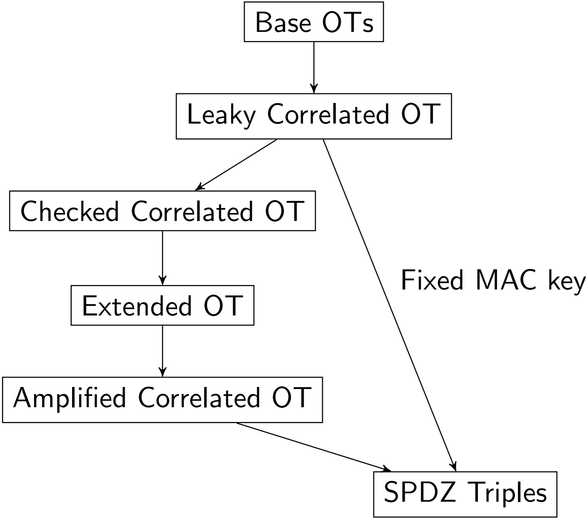
Protocol

1. Pairwise amplified OT to get secret-shared triple
2. Pairwise leaky correlated OT to get secret-shared MACs
3. Improved SPDZ sacrifice (error detection)

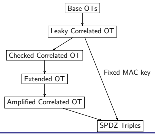
Attacks

- Cheating in amplified OT:
  Erased by amplification or fail error detection
- Cheating in leaky correlated OT:
  Some bits of MAC key are leaked. Not an issue because complete leakage succeeds only with negligible probability
- Use different inputs for different parties: fail error detectin

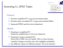
Toolchain



Toolchain



Base OTs
Leaky Correlated OT
Checked Correlated OT
Extended OT
Amplified Correlated OT
Fixed MAC key
SPDZ Triples



Toolchain

Field Multiplication from Tensor Product

$$(1, X, X^2, \dots) \cdot (\mathbf{x} \otimes \mathbf{y}) \cdot (1, X, X^2, \dots)^\top =$$

$$\begin{pmatrix} 1 & X & X^2 & \dots \end{pmatrix} \cdot \begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 & \\ x_2y_1 & x_2y_2 & x_2y_3 & \dots \\ x_3y_1 & x_3y_2 & x_3y_3 & \\ & \vdots & & \ddots \end{pmatrix} \cdot \begin{pmatrix} 1 \\ X \\ X^2 \\ \vdots \end{pmatrix}$$

$$= x_1y_1 + (x_1y_2 + x_2y_1) \cdot X + (x_1y_3 + x_2y_2 + x_3y_1) \cdot X^2 + \dots$$

$$= \mathbf{x} \cdot \mathbf{y}$$

Hashing Step

Field Multiplication from Tensor Product

$$(1, X, X^2, \dots) \cdot (\mathbf{x} \otimes \mathbf{y}) \cdot (1, X, X^2, \dots)^\top =$$

$$\begin{pmatrix} 1 & X & X^2 & \dots \end{pmatrix} \cdot \begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 & \\ x_2y_1 & x_2y_2 & x_2y_3 & \dots \\ x_3y_1 & x_3y_2 & x_3y_3 & \\ & \vdots & & \ddots \end{pmatrix} \cdot \begin{pmatrix} 1 \\ X \\ X^2 \\ \vdots \end{pmatrix}$$

$$= x_1y_1 + (x_1y_2 + x_2y_1) \cdot X + (x_1y_3 + x_2y_2 + x_3y_1) \cdot X^2 + \dots$$

$$= \mathbf{x} \cdot \mathbf{y}$$

Extended random OT: j-th input is $H(\mathbf{q}_j)$ and $H(\mathbf{q}_j + \mathbf{x})$

$$H(\mathbf{q}_j) = H(\mathbf{t}_j + \mathbf{x} * \mathbf{z}_j)$$
$$H(\mathbf{q}_j + \mathbf{x}) = H(\mathbf{t}_j + \mathbf{x} * \overline{\mathbf{z}_j})$$

Theorem (Sufficient)

If the check passes with probability 2^{-m}, then $\kappa - m$ bits of either $\mathbf{x} * \mathbf{z}_j$ or $\mathbf{x} * \overline{\mathbf{z}_j}$ remain unknown to the sender of the base OT / receiver of the extended OT.