Private Function Evaluation - From Functions to Data to Code

Thomas Schneider

Based on joint works with Ágnes Kiss, Daniel Günther, Masaud Y. Alhassan, N. Asokan (Aalto U), Jian Liu (UC Berkeley), Masoud Naderpour (U Helsinki)

TPMPC, June 2019

2PC  
3PC  
MPC  

\( f \rightarrow p \rightarrow c \)
Secure Function Evaluation (SFE)
Secure Function Evaluation of Boolean Circuits

Garbling (à la Yao’s Garbled Circuit Protocol) or
Secret Sharing (à la Goldreich-Micali-Wigderson Protocol)
Private Function Evaluation (PFE)
1. Private Function Evaluation of Boolean Circuits

2. Private Function Evaluation of Decision Trees
Private Function Evaluation of Boolean Circuits
Applications of PFE of Boolean Circuits

Solvency verification

Smart metering

Private databases

Insurance rate & credit risk assessment
Challenges – Hiding the Circuit

• Public:
  • Number of inputs $u$
  • Number of outputs $v$
  • Number of gates $k$

• Private:
  • Functionality of gates
  • Topology of circuit

$u = 4$
$v = 1$
$k = 4$
There exists a Boolean circuit $UC$ of size $\Theta(n \log n)$ s.t. for any Boolean function $f$ of size $n$ $UC$ can be programmed to compute $f$.

Leslie G. Valiant
1976
Universal Circuit (UC)

There exists a Boolean circuit $UC$ of size $\Theta(n \log n)$ s.t. for any Boolean function $f$ of size $n$ there exists a programming $p$ such that for any input $x$: $UC(p, x) = f(x)$.
PFE of Boolean Circuits
PFE of Boolean Circuits via SFE of a UC

\[ f(x) = UC(p, x) \]
Further Applications of UCs beyond PFE

- Obfuscation
- Attribute-based Encryption
- Batch Execution MPC
- Adaptively Secure MPC
Universal circuit $UC$ Programming bits $p$
# Existing UC Constructions


<table>
<thead>
<tr>
<th></th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>$5n \log n$</td>
<td>$4.75n \log n$</td>
</tr>
<tr>
<td><strong>Depth</strong></td>
<td>$3n$</td>
<td>$3.75n$</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>
Valiant’s UC Construction

$n \quad C \text{ size} \leq n$

Graph $G_C$
Valiant’s UC Construction

Universal graph $UG$ $\rightarrow$ Embedding $E$

Universal circuit $UC$

Graph $G_C$

$C$ size $\leq n$
Valiant’s UC Construction

Universal graph $UG$ → Embedding $E$

Universal circuit $UC$

$n$

$C$ size $\leq n$

Graph $G_C$
Valiant’s UC Construction

Universal graph $UG$ $\rightarrow$ Embedding $E$

Universal circuit $UC$

$G_C$ $\subseteq$ $n$

$C$ size $\leq n$

$UG$ $\rightarrow$ $G_C$
Valiant’s UC Construction

Universal graph $UG$ → Embedding $E$

Universal circuit $UC$

Graph $G_C$

$C$ size $\leq n$

$n$

Programming bits $p$
2-way Recursive UG Construction

$UG_n \rightarrow UG_{n/2}^1 \rightarrow UG_{n/4}^{11} \rightarrow UG_{n/4}^{12} \rightarrow \ldots \rightarrow UG_{n/4}^{21} \rightarrow UG_{n/4}^{22} \rightarrow UG_{n/2} \rightarrow UG_n$
2-way Recursive UG Construction

$UG_n 
\begin{align*}
UG^1_{n/2} & \rightarrow UG^{11}_{n/4} \\
& \rightarrow UG^{111}_{n/8} \\
& \quad \rightarrow UG^{1111}_{n/8} \\
& \quad \quad \rightarrow UG^{11111}_{n/8}
\end{align*}

$UG^2_{n/2} 
\begin{align*}
UG^2_{n/2} & \rightarrow UG^{12}_{n/4} \\
& \rightarrow UG^{121}_{n/8} \\
& \quad \rightarrow UG^{1211}_{n/8} \\
& \quad \quad \rightarrow UG^{12111}_{n/8}
\end{align*}$
A „Small„ Example

\[ u = 25 \]

\[ k = 56 \]

\[ v = 1 \]

835 nodes / 869 AND gates

\[ f \]

\[ UC \]
Existing UC Constructions

<table>
<thead>
<tr>
<th></th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
<th>[KS08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>$5n \log n$</td>
<td>$4.75n \log n$</td>
<td>$1.5n \log^2 n + 2n \log n$</td>
</tr>
<tr>
<td>Depth</td>
<td>$3n$</td>
<td>$3.75n$</td>
<td>$n \log n$</td>
</tr>
<tr>
<td>Code</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>

Existing UC Constructions

<table>
<thead>
<tr>
<th></th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
<th>[KS08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>$5n \log n$</td>
<td>$4.75n \log n$</td>
<td>$1.5n \log^2 n + 2n \log n$</td>
</tr>
<tr>
<td>Depth</td>
<td>$3n$</td>
<td>$3.75n$</td>
<td>$n \log n$</td>
</tr>
<tr>
<td>Code</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Comparison

Input circuit size $n = 1070$
Existing UC Constructions

<table>
<thead>
<tr>
<th>Year</th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
<th>[KS08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>$5n \log n$</td>
<td>$4.75n \log n$</td>
<td>$1.5n \log^2 n + 2n \log n$</td>
</tr>
<tr>
<td>Depth</td>
<td>$3n$</td>
<td>$3.75n$</td>
<td>$n \log n$</td>
</tr>
<tr>
<td>Code</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>


[KS08] 2008

[KS16] 2016

2-way Recursive UG Construction [Val76]
4-way Recursive UG Construction [Val76]

$UG_n$
4-way Modular Embedding Algorithm

Task 1: Block embedding
Task 2: Recursion point embedding
Concrete Size of UCs

**Blue: Improvement of 4-way UC over 2-way UC**

$$\text{Improvement in percent}$$

<table>
<thead>
<tr>
<th>Input circuit size</th>
<th>AES-128</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 38 518</td>
<td>2.36 mio</td>
<td>14.8 mio</td>
</tr>
<tr>
<td>n = 201 206</td>
<td>2.37 mio</td>
<td>14.5 mio</td>
</tr>
</tbody>
</table>

Maximum: $$\frac{5}{4.75} - 100\% = 5.3\%$$

2-way UC is better
## Existing UC Constructions

<table>
<thead>
<tr>
<th>Year</th>
<th>Construction</th>
<th>Size</th>
<th>Depth</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1976</td>
<td>[Val76]</td>
<td>$5n \log n$</td>
<td>$3n$</td>
<td>✅</td>
</tr>
<tr>
<td>2008</td>
<td>[Val76]</td>
<td>$4.75n \log n$</td>
<td>$3.75n$</td>
<td>✅</td>
</tr>
<tr>
<td>2008</td>
<td>[KS08]</td>
<td>$1.5n \log^2 n + 2n \log n$</td>
<td>$n \log n$</td>
<td>✅</td>
</tr>
<tr>
<td>2016</td>
<td>[KS16]</td>
<td>$4.75n \log n$</td>
<td>$3.75n$</td>
<td>✗</td>
</tr>
<tr>
<td>2017</td>
<td>[GKS17]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Hybrid UC

UC for size $n$

?  
4-way split $<> 2$-way split

⇒ At each recursion step: choose smallest construction
Concrete Size of UCs – Hybrid UC

Green: Improvement of hybrid UC over 2-way UC
Blue: Improvement of 4-way UC over 2-way UC

Hybrid UC is better than both UCs

Maximum: \( \frac{5}{4.75} - 100\% = 5.3\% \)
### Existing UC Constructions

<table>
<thead>
<tr>
<th></th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
<th>[KS08]</th>
<th>[GKS17] Hybrid(2, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>5(n \log n)</td>
<td>4.75(n \log n)</td>
<td>1.5(n \log^2 n) (+ 2n \log n)</td>
<td>4.75(n \log n) (+ 4.5n \log n)</td>
</tr>
<tr>
<td><strong>Depth</strong></td>
<td>3(n)</td>
<td>3.75(n) (+ 3.5n)</td>
<td>(n \log n)</td>
<td>3.75(n) (+ 3.5n)</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
</tr>
</tbody>
</table>

[Val76] Valiant's original construction.


Concrete Size of UCs – Improvement of [ZYSL18]

Red: Improvement of hybrid UC with [ZYSL18] 4-way UC over 2-way UC
Yellow: Improvement of [ZYSL18] 4-way UC over 2-way UC
Green: Improvement of hybrid UC over 2-way UC
Blue: Improvement of 4-way UC over 2-way UC

Maximum: \( \frac{5}{4.5} - 100\% = 11.1\% \)
Maximum: \( \frac{5}{4.75} - 100\% = 5.3\% \)
## Existing UC Constructions

<table>
<thead>
<tr>
<th>Year</th>
<th>[Val76] 2-way</th>
<th>[Val76] 4-way</th>
<th>[KS08]</th>
<th>[GKS17] Hybrid(2, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1976</td>
<td>$5n \log n$</td>
<td>$4.75n \log n$</td>
<td>$1.5n \log^2 n + 2n \log n$</td>
<td>$4.75n \log n$ + $4.5n \log n$</td>
</tr>
<tr>
<td>2008</td>
<td>$4.5n \log n$</td>
<td>$3.75n$</td>
<td>$n \log n$</td>
<td>$3.75n$</td>
</tr>
<tr>
<td>2016</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2018</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>[Val76]</th>
<th>[KS08]</th>
<th>[GKS17]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**+ Scalability**

Scalable 4-way UC Implementation

Max. Memory for UC Generation (MB)

- 32 GB
- 16 GB
- 8 GB
- 2 GB
- 1 GB

Input circuit size $n$

$O(n \log n)$

$O(n)$

Generation runtime (ms)

$O(n \log n)$

$O(n \log n)$

Input circuit size $n$
UC Implementation

UC Implementation

\[ \text{[KS16]} \]

\[ C \text{ size} \leq n \leftrightarrow C_0 \leftrightarrow f \]

\[ \text{[KS16]} \] Á. Kiss, T. Schneider: Valiant's Universal Circuit is Practical. In EUROCRIPT’16.
$C \text{ size } \leq n$

Graph $G_C$
UC Implementation

Universal graph $UG$ → Edge-embedding $E$

Universal circuit $UC$ → Programming bits $p$

$n$ → $C$ size $\leq n$ → Graph $G_C$
UC Implementation

Universal circuit $UC$ Programming bits $p$

$C \text{ size } \leq n$

$n$

We have code!

Code at https://encrypto.de/code/UC
Experimental Results – UC Compiler (one-time expense)

- **Input circuit size**
  - 9

- **Generation time (ms)**
  - Hybrid, 4-way with [ZYZL18]: 12 s
  - Hybrid, 4-way with [Val76]: 14 s
  - Total: 12 s to 2 min
Implementation of PFE via UC

\[ UC(p, x) = f(x) \]

UC Compiler

Universal Circuit \( UC \)

Programming bits \( p \)

\[ C \text{ size} \leq n \]

\[ C_0 \leftarrow f \]

Runtime and Communication for PFE of Boolean Circuits

LAN: 10 Gbps, 1ms RTT
WAN: 100Mbps, 100ms RTT
Conclusions for PFE of Boolean Circuits

• Universal Circuits are a competitive solution for PFE of Boolean Circuits

• Performance of UC-based PFE (using Yao’s GC in ABY):
  – AES \( n = 3\,8518 \): 2s in LAN; 11s in WAN
  – \( n = 1\,000\,000 \): 1.3 min in LAN; 5.9 mins in WAN

• Extending secure computation frameworks for PFE with UCs is simple
  – Simple adapter for UC format (similar to Fairplay’s SHDL)
  – Code at https://encrypto.de/code/UC
1. Private Function Evaluation of Boolean Circuits

2. Private Function Evaluation of Decision Trees
Decision Trees (DTs)

Decision node $i$: comparison between one of the input features $x_j$ and thresholds $y_i$

Leaf or classification nodes: store classification result

[Bryant91]: Some functionalities such as multiplication require exponential size DTs
Private Function Evaluation of Decision Trees
Applications of PFE of Decision Trees

Software diagnostics

Medical diagnostics

Text classification

Malware classification
Challenges – Hiding the Decision Tree

- **Public:**
  - Number of input features $n$
  - Number of decision nodes $m \geq n$
  - (Depth of decision tree $d$)

- **Private:**
  - Which feature is compared at the decision node and the threshold
  - Comparison result
  - Tree topology
    - length of specific evaluated path

$n = 2$ features $x_1, x_2$
m = 2 decision nodes
$(d = 2)$ depth
1. Oblivious selection of features to compare with thresholds

2. Oblivious comparison with thresholds

3. Oblivious path evaluation

H: Homomorphic Encryption

\[ Sel_H \]

\[ Comp_H \]

\[ Path_H \]

G: Garbling

\[ Sel_G \]

\[ Comp_G \]

\[ Path_G \]

Homomorphic Encryption-based Approaches (H)

- **Oblivious selection** $Sel_H$:
  - Alice sends encrypted inputs to Hospital, Hospital obliviously selects and blinds, and sends back to Alice who decrypts
  - HE instantiated with DGK or Paillier encryption; both parties have additive sharing of the obliviously selected attribute

- **Oblivious comparison** $Comp_H$ (can also include oblivious selection $Sel_H$):
  - DGK comparison [DGK07]:
    - Hospital evaluates a linear function on Alice’s bitwise encrypted inputs that results in 0 iff comparison is true
    - Hospital blinds result with random bit (Hospital’s share of the comparison output) before sending to Alice
    - Alice sets her share to 1 if encryption of 0, and to 0 otherwise
  - Instantiated with lifted EC-ElGamal, parties learn an XOR secret sharing of the comparison

- **Oblivious path evaluation** $Path_H$:
  - Set edge costs: the cost of the left/right edge is 1/0 if comparison is true, otherwise 0/1
  - Hospital calculates path costs (sum along all evaluation paths) under encryption
  - Alice finds successful path whose path cost decrypts to 0 & decrypts classification value
  - Instantiated with lifted EC-ElGamal

Garbling-based Approaches (G)

- **Oblivious selection \( Sel_G \):**
  - Evaluate selection network of size \( O(m \log m) \) using Yao’s GC (Hospital provides selection, Alice provides attributes)
  - Retrieve keys or XOR shares of the result

- **Oblivious comparison \( Comp_G \):**
  - Evaluate comparison circuit using Yao’s GC
  - Retrieve keys or XOR shares of the result

- **Oblivious path evaluation \( Path_G \):**
  - Garbled decision tree evaluation, similar to Yao’s GC
    - Hospital chooses a random key \( \Delta_i \) for each decision node and two keys \( k^L_i, k^R_i \) for each edge; then garbles decision node
    - Alice evaluates the tree using the keys \( k^L_i/R_i \) from the oblivious comparison
  - Efficient but requires depth-padding

- **Easy to secure against malicious Alice using maliciously secure OT extension**
### Complexity Comparison for PFE of Decision Trees

<table>
<thead>
<tr>
<th></th>
<th>Homomorphic Encryption [TaiMZC17]</th>
<th>Garbling [BFKLSS09]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computation</strong></td>
<td>$0(\sigma m)$ homomorphic public-key</td>
<td>$O(\sigma \tilde{m} \log \tilde{m})$ symmetric</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td>$0(\sigma m)$ homomorphic public-key</td>
<td>$0(\sigma \tilde{m} \log \tilde{m})$ symmetric</td>
</tr>
</tbody>
</table>

- $\sigma$: bitlength of attributes and thresholds
- $m$: number of decision nodes
- $\tilde{m}$: number of depth-padded decision nodes
- $d$: depth of decision tree


Our Systematic Comparison

<table>
<thead>
<tr>
<th></th>
<th>$Path_G$</th>
<th>$Path_H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Sel_G + Comp_G$</td>
<td>GGG [BFKLSS09]</td>
<td>GGH (new [KMLAS19])</td>
</tr>
<tr>
<td>$Sel_H + Comp_G$</td>
<td>HGG [BFKLSS09]</td>
<td>HGH (new [KMLAS19])</td>
</tr>
<tr>
<td>$Sel_H + Comp_H$</td>
<td>HHG (new but inefficient)</td>
<td>HHH [TMZC17]</td>
</tr>
<tr>
<td>$Sel_G + Comp_H$</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
### Example Decision Trees from the UCI Repository

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#input features</th>
<th>depth</th>
<th>#decision nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>3</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>wine</td>
<td>7</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>linnerud</td>
<td>3</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>breast cancer</td>
<td>12</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>digits</td>
<td>47</td>
<td>15</td>
<td>168</td>
</tr>
<tr>
<td>diabetes</td>
<td>10</td>
<td>28</td>
<td>393</td>
</tr>
<tr>
<td>boston</td>
<td>13</td>
<td>30</td>
<td>425</td>
</tr>
</tbody>
</table>
Runtimes

**Online runtime in ms**

- (HHH) [BFK+09]
- HHH [TMZC17]
- HGH
- GGG [BFK+09]
- GGH

**Total runtime in ms**

- (HHH) [BFK+09]
- HHH [TMZC17]
- HGH
- GGG [BFK+09]
- GGH
Communication

Online communication in KBytes

Total communication in KBytes
Tradeoffs: Setup & Online

Tradeoffs for our largest dataset Boston

Unfilled forms: setup phase
Filled forms: online phase
Tradeoffs: Total

Tradeoffs for our largest dataset Boston

Total runtime in ms vs Total communication in KBytes

- HHH [TMZC17]
- HGH
- GGH
- (HHG) [BFK+09]
- HGG [BFK+09]
- GGG [BFK+09]
Conclusions for PFE of Decision Trees

- **GGG [BFKLSS09]:** Lowest *online runtime and online communication*
- **HHH [TMZC17]:** Lowest *total communication*
- **GGH [KMLAS19]:** Lowest *total runtime*
- **HGH [KMLAS19]:** Tradeoff between HHH and GGH

Code available soon at [https://encrypto.de/code/PDTE](https://encrypto.de/code/PDTE)
Thanks for your attention!

Questions?

Contact: https://encrypto.de