Applying the Trustworthy Remote Entity to Privacy-Preserving Multiparty Computation
Requirements and Criteria for Large-Scale Applications

Robin Ankele\textsuperscript{1}, Kubilay Küçük\textsuperscript{1}, Andrew Martin\textsuperscript{1}, Andrew Simpson\textsuperscript{1}, Andrew Paverd\textsuperscript{2}

\textsuperscript{1}University of Oxford, UK \hspace{2cm} \textsuperscript{2}Aalto University, FI

July 19, 2016
1. Paper Outline

2. Secure Multiparty Computation

3. Trustworthy Remote Entity

4. Use Cases

5. Requirements and Criteria
Requirements and Criteria for Large-Scale Multiparty Applications

1 Secure Multiparty Computation
2 Trusted Computing
Secure Multiparty Computation

1. Paper Outline

2. Secure Multiparty Computation

3. Trustworthy Remote Entity

4. Use Cases

5. Requirements and Criteria
Secure Multiparty Computation Problem

Set of parties $P_1, \ldots, P_n$, each with their private input $x_1, \ldots, x_n$, jointly computing a function $f(x_1, \ldots, x_n) = y$. 
Secure Multiparty Computation Problem

Set of parties $P_1, \ldots, P_n$, each with their private input $x_1, \ldots, x_n$, jointly computing a function $f(x_1, \ldots, x_n) = y$

- Mutually Distrustful Parties
- Distributed
- Controlled/Observed by Adversaries
Secure Multiparty Computation Problem
Set of parties $P_1, \ldots, P_n$, each with their private input $x_1, \ldots, x_n$, jointly computing a function $f(x_1, \ldots, x_n) = y$

- Mutually Distrustful Parties
- Distributed
- Controlled/Observed by Adversaries
Secure Multiparty Computation

Secure Multiparty Computation Problem
Set of parties $P_1, \ldots, P_n$, each with their private input $x_1, \ldots, x_n$, jointly computing a function $f(x_1, \ldots, x_n) = y$

- Mutually Distrustful Parties
- Distributed
- Controlled/Observed by Adversaries
Secure Multiparty Computation Problem

Set of parties $P_1, \ldots, P_n$, each with their private input $x_1, \ldots, x_n$, jointly computing a function $f(x_1, \ldots, x_n) = y$

- Mutually Distrustful Parties
- Distributed
- Controlled/Observed by Adversaries
sMPC Protocols

Theoretical Implementations
- Yao[8], GMW[6], BMR[1] and BGW[3]

Practical Implementations
- VIFF[5], TASTY[7], Sharemind[4] and Fairplay[2]
sMPC Protocols

Theoretical Implementations
- Yao[8], GMW[6], BMR[1] and BGW[3]

Practical Implementations
- VIFF[5], TASTY[7], Sharemind[4] and Fairplay[2]
Problems with Current sMPC Implementations

Efficiency

- Communication overhead
- Use of expensive operations
- Parameters for FHE

(Output) Privacy

- sMPC schemes are only concerned about (input) privacy
- Result of function itself reveals information
Problems with Current sMPC Implementations

Efficiency

- Communication overhead
- Use of expensive operations
- Parameters for FHE

(Output) Privacy

- sMPC schemes are only concerned about (input) privacy
- Result of function itself reveals information
Problems with Current sMPC Implementations

Example

- Computing the average salaries of $P_1 \ldots P_n$
- With $(n - 1)$ colluding parties — reveals salary of $P_n$
- Two-party case — knowledge of average and own salary is sufficient

Achieving (Output) Privacy

- Privacy models and privacy budget
Problems with Current sMPC Implementations

Example

- Computing the average salaries of $P_1 \ldots P_n$
- With $(n - 1)$ colluding parties — reveals salary of $P_n$
- Two-party case — knowledge of average and own salary is sufficient

Achieving (Output) Privacy

- Privacy models and privacy budget
1. Paper Outline

2. Secure Multiparty Computation

3. Trustworthy Remote Entity

4. Use Cases

5. Requirements and Criteria
Indistinguishability

\[ \Pr[\text{Real} = 1] - \Pr[\text{Ideal} = 1] < \mu \]
### TTP vs. TRE

<table>
<thead>
<tr>
<th>TTP</th>
<th>TRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>blindly trusted</td>
<td>verifiable</td>
</tr>
<tr>
<td>no trust est.</td>
<td>attestation-based</td>
</tr>
<tr>
<td>-</td>
<td>TPM\textsuperscript{1}, TEE\textsuperscript{2}</td>
</tr>
</tbody>
</table>

### Privacy

- TRE for computational privacy
- Privacy-preserving algorithms on TRE for output privacy

\textsuperscript{1} Trusted Platform Module
\textsuperscript{2} Trusted Execution Environment
TTP vs. TRE

<table>
<thead>
<tr>
<th>TTP</th>
<th>TRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>blindly trusted</td>
<td>verifiable</td>
</tr>
<tr>
<td>no trust est.</td>
<td>attestation-based</td>
</tr>
<tr>
<td>-</td>
<td>TPM(^1), TEE(^2)</td>
</tr>
</tbody>
</table>

Privacy

- TRE for computational privacy
- Privacy-preserving algorithms on TRE for output privacy

\(^1\) Trusted Platform Module  
\(^2\) Trusted Execution Environment
Software Guard Extensions

- Instruction set extensions
- Protects user-space applications from higher-privileged operations (e.g. operating system, drivers)
- Hardware enforced isolation of user-space memory pages (code + data) in so called *enclaves*
- Secure sealing and binding of data pages
- Local and remote attestation
Software Guard Extensions

- Instruction set extensions
- Protects user-space applications from higher-privileged operations (e.g. operating system, drivers)
- Hardware enforced isolation of user-space memory pages (code + data) in so called *enclaves*
- Secure sealing and binding of data pages
- Local and remote attestation
Software Guard Extensions

- Instruction set extensions
- Protects user-space applications from higher-privileged operations (e.g. operating system, drivers)
- Hardware enforced isolation of user-space memory pages (code + data) in so called enclaves
- Secure sealing and binding of data pages
- Local and remote attestation
Software Guard Extensions

- Instruction set extensions
- Protects user-space applications from higher-privileged operations (e.g. operating system, drivers)
- Hardware enforced isolation of user-space memory pages (code + data) in so called enclaves
- Secure sealing and binding of data pages
- Local and remote attestation
Software Guard Extensions

- Instruction set extensions
- Protects user-space applications from higher-privileged operations (e.g., operating system, drivers)
- Hardware enforced isolation of user-space memory pages (code + data) in so-called enclaves
- Secure sealing and binding of data pages
- Local and remote attestation
Benefits of a SGX-based TRE

- Executing on the main CPU
- Reducing the TCB\(^3\) size

\(^3\)Trusted Computing Base
Use Cases

1. Paper Outline

2. Secure Multiparty Computation

3. Trustworthy Remote Entity

4. Use Cases

5. Requirements and Criteria
Use Cases

- Smart Grid
- Road Pricing
- Privacy-Preserving Data Mining
- Auctions
- Elections
- Private Set Intersections
Problem 1. (Network Monitoring)

- Time-of-use, energy usage
- Measurements of energy consumption in intervals (e.g. half-hourly; hourly; daily)
- Non-intrusive load monitoring techniques

Analysis.

- Highly-time dependent energy measurements
- Anonymity of energy consumer
Problem 2.  

- Unpredictable consumer behaviour or unplanned drops or rises in energy consumption  
- Broadcasts a request for reduction of energy  
- Bidding Protocol

Analysis.

- Privacy of bids  
- Anonymity of bidder  
- Non-repudiation of bidder
Requirements and Criteria

1. Paper Outline

2. Secure Multiparty Computation

3. Trustworthy Remote Entity

4. Use Cases

5. Requirements and Criteria
Requirement 1.
We require privacy-preservation of the input and output. We refer to the former as input privacy and to the latter as output privacy.
Requirement 2.
We require *anonymity* of parties participating in the protocol. However, we also require *non-repudiation* of committed data values.
Requirement 3.
We require correctness of the function evaluation, in the presence of malicious parties.
Requirement 4.
We require guaranteed output delivery of the function evaluation to all trustworthy and malicious parties.
Requirement 5.
The parties should be able to maintain a privacy budget devoting the level of trust in the application.
Requirement 6.
We require an *efficient* and *scalable* implementation of the privacy-preserving multi-party protocol.
Evaluation Criteria of Large-Scale Prototypes

Evaluation Criteria

- Computational Models
  - Communication model, adversarial model
- General Criteria
  - Communication costs, computational costs
- Large-Scale Criteria
  - Parallelisation, precomputation, hardware-assisted design
  - Communication locality, load balancing
- Additional Metrics
  - Reusability, interoperability and minimal design
  - Universal composability
  - Privacy impact of function
Evaluation Criteria of Large-Scale Prototypes

Evaluation Criteria

- **Computational Models**
  - Communication model, adversarial model

- **General Criteria**
  - Communication costs, computational costs

- **Large-Scale Criteria**
  - Parallelisation, precomputation, hardware-assisted design
  - Communication locality, load balancing

- **Additional Metrics**
  - Reusability, interoperability and minimal design
  - Universal composability
  - Privacy impact of function
Evaluation Criteria of Large-Scale Prototypes

Evaluation Criteria

▶ Computational Models
  ▶ Communication model, adversarial model

▶ General Criteria
  ▶ Communication costs, computational costs

▶ Large-Scale Criteria
  ▶ Parallelisation, precomputation, hardware-assisted design
  ▶ Communication locality, load balancing

▶ Additional Metrics
  ▶ Reusability, interoperability and minimal design
  ▶ Universal composability
  ▶ Privacy impact of function
Evaluation Criteria of Large-Scale Prototypes

Evaluation Criteria

▶ Computational Models
  ▶ Communication model, adversarial model

▶ General Criteria
  ▶ Communication costs, computational costs

▶ Large-Scale Criteria
  ▶ Parallelisation, precomputation, hardware-assisted design
  ▶ Communication locality, load balancing

▶ Additional Metrics
  ▶ Reusability, interoperability and minimal design
  ▶ Universal composability
  ▶ Privacy impact of function
Conclusions

Use Cases

- Smart Grid, Road Pricing, PPDM, Elections, Auctions, Set Intersection

Requirements and Criteria

- Large-scale distributed multi-party applications

Application to the Trustworthy Remote Entity

- Trusted Computing based
- Efficient and secure (i.e. verifiable trustworthiness)
- Computational and output privacy
Conclusions

Use Cases
- Smart Grid, Road Pricing, PPDM, Elections, Auctions, Set Intersection

Requirements and Criteria
- Large-scale distributed multi-party applications

Application to the Trustworthy Remote Entity
- Trusted Computing based
- Efficient and secure (i.e. verifiable trustworthiness)
- Computational and output privacy
Conclusions

Use Cases

- Smart Grid, Road Pricing, PPDM, Elections, Auctions, Set Intersection

Requirements and Criteria

- Large-scale distributed multi-party applications

Application to the Trustworthy Remote Entity

- Trusted Computing based
- Efficient and secure (i.e. verifiable trustworthiness)
- Computational and output privacy
Future Work

Planned Work

- Implementation of SGX based TRE using Intel SGX SDK
- Selection of suitable privacy algorithms / with privacy budget
- Scalability of SGX TRE approach
- Performance benchmarking
Applying the Trustworthy Remote Entity to Privacy-Preserving Multiparty Computation
Requirements and Criteria for Large-Scale Applications

Robin Ankele\textsuperscript{1}, Kubilay Küçük\textsuperscript{1}, Andrew Martin\textsuperscript{1}, Andrew Simpson\textsuperscript{1}, Andrew Paverd\textsuperscript{2}

\textsuperscript{1}University of Oxford, UK \quad \textsuperscript{2}Aalto University, FI

July 19, 2016
D. Beaver, S. Micali, and P. Rogaway.  
The Round Complexity of Secure Protocols.  

A. Ben-David, N. Nisan, and B. Pinkas.  
FairplayMP: A System for Secure Multi-party Computation.  

M. Ben-Or, S. Goldwasser, and A. Wigderson.  
Completeness Theorems for Non-cryptographic Fault-tolerant Distributed Computation.  

D. Bogdanov, S. Laur, and J. Willemson.  
I. Damgård, M. Geisler, M. Krøigaard, and J. Nielsen.


O. Goldreich, S. Micali, and A. Wigderson.
How to Play ANY Mental Game.

TASTY: Tool for Automating Secure Two-party Computations.

A. Yao.
Protocols for Secure Computations.
Backup Slides
Properties of a sMPC Protocol

- (Input) Privacy
- Correctness
- Independence of Inputs
- Fairness
- Guaranteed Output Delivery
Adversarial Model

Semi-Honest (i.e. passive)
- Follows protocol
- Infer information of other participants

Covert
- Deviates from the protocol, unless a high probability of detection

Malicious (i.e. active)
- Arbitrary deviates from the protocol
Trustworthy Remote Entity

Ideal World

- Trusted Third Party
- Must be blindly trusted

Trustworthy Remote Entity

- Introduced by Paverd
- Applications in the smart grid scenario
- Trusted Computing based
Trustworthy Remote Entity

Features of a TRE

▶ Isolation of code and data
▶ Secure storage
▶ Local and remote attestation

Communication Benefits

▶ Privacy-Preserving Bi-Directional Communication
▶ FSA\(^4\) Protocol

\(^4\) Finite State Attestation
Problem 3. (Network Monitoring)

- Time-of-use, energy usage
- Measurements of energy consumption in intervals (e.g. half-hourly; hourly; daily)
- Non-intrusive load monitoring techniques

Analysis.

- Highly-time dependent energy measurements
- Anonymity of energy consumer
Problem 4. (Billing)

- Time-of-use, dynamic billing
- Calculation of bill by potentially malicious energy consumer

Analysis.

- Time-sensitive billing information
- Anonymity of energy consumer
- Non-repudiation of energy consumer
Problem 5. (Demand control)

- Unpredictable consumer behaviour or unplanned drops or rises in energy consumption
- Broadcasts a request for reduction of energy
- Bidding Protocol

Analysis.

- Privacy of bids
- Anonymity of bidder
- Non-repudiation of bidder
Problem 6. 

- Scanning identifiers at toll stations (e.g. license plate, on-board-unit)
- Analyse road usage, prevent traffic jams and congestions, route planners, vehicle types
- Real time tracking and the detection of movement patterns

Analysis.

- Location and time of road usage
- Anonymity of vehicle owner
Problem 7. (Billing)

- New road billing schemes (e.g. distance, time, pay-as-you-drive insurance/road tax)
- Submission of detailed traffic information (e.g. location of vehicle, time of use)

Analysis.

- Location and time of road usage
- Anonymity of vehicle owner
- Non-repudiation of vehicle owner
Problem 8. (Genomics)

- Studies behaviour of diseases
- Genomic data sets (genome information), patient data sets (personal information)
- Genetic data can reveal kinship, ethnicity or the location of outbreak

Analysis.

- Anonymity of patients
- Privacy of location and ethnicity of patients
Problem 9. (Intrusion Detection)

- Database containing metrics of intrusion behaviour — comparing to other companies database
- Adversary’s behaviour of recent break-in — comparing to profile databases
- Monitoring employee’s behaviour to detect leakage of sensitive information to outsiders

Analysis.

- Privacy of databases or sensitive data patterns
- Privacy of employee
Problem 10. (Auctions)

- Selection of auction winner
- Learn input of all participating parties
- Obfuscate participation and content

Analysis.

- Privacy of bids
- Anonymity of bidder
Problem 11. (Voting)

- Majority of votes
- Learn vote content and eligibility to vote
- Correctness, non-repudiation and anonymity of voter and content

Analysis.

- Privacy of vote
- Non-repudiation of vote
- Anonymity of voter
Problem 12. (Set Intersections)

- Intersection of private databases
- Privacy-preserving comparison of databases

Analysis.

- Privacy of database content
Evaluation Criteria of Large-Scale Prototypes

Evaluation Criteria

▶ Computational Models
▶ General Criteria
▶ Large-Scale Criteria
▶ Additional Metrics
Communication model

- Synchronous, asynchronous
- Upper bound on communication time
- Correctness, guaranteed output delivery

Adversarial model

- Semi-honest, covert, malicious
- Static, adaptive
- Fault tolerance
General Criteria

Communication Costs
- Number and size of messages \( m, \ell \)
- Number of participants \( n \)
- Communication rounds \( d \)

Computational Costs
- Time complexity \( \mathcal{O}(n) \)
- Memory complexity \( \mathcal{O}(m \cdot \ell) \)
Parallelisation and Precomputation

- Execute operations of several parties simultaneously
- Online/offline phase
- Precomputation (e.g. attestation, key exchange)

Hardware-assisted design

- SIMD\(^5\), SHA-EXT\(^6\), SGX

\(^5\)Single Instruction Multiple Data (e.g. Intel\(^\text{®}\) SSE, AVX instructions)
\(^6\)Intel\(^\text{®}\) SHA extensions
Large-Scale Criteria cont.

Communication Locality
- Number of nodes each party communicates with

Load Balancing
- Balance of computational tasks of each party
Additional Metrics

- Reusability of parts of the protocol
- Interoperability and minimal design
- Universal composability
- Privacy impact of function