LPDA-EC: A Lightweight Privacy-Preserving Data Aggregation Scheme for Edge Computing

Jiale Zhang, Yanchao Zhao, Jie Wu & Bing Chen

@ Nanjing University of Aeronautics and Astronautics & Temple University

IEEE MASS 2018 Chengdu, China
Simple application scenario: **smart grid**

- **Users collect the sensitive data**
- **Then, forward them to the data center**
- **Making the intelligent decisions**
Traditional data transmission

- Communication overhead
- Adversary can eavesdrop the channel
- System entities may not fully trusted
- User’s private data may leakage

How to efficiently transmit the data while protecting user’s privacy?
PPDA solution

- PPDA: Privacy-preserving data aggregation
  - Cryptographic scheme to protect the data privacy
  - Signature scheme to ensure the integrity

Reducing the communication overhead!
So what’s the problem?

- We can ensure the user’s privacy — cryptographic
- Data can be aggregated — homomorphic
- The data integrity can be guaranteed — signature
- Why can’t we just use it?
Problem statement

- Two small wrinkles:
  - Complex signature and verification operations

Diagram:

- User1 reports $\text{Sign}_1$
- User2 reports $\text{Sign}_2$
- User $\text{User}_n$ reports $\text{Sign}_n$
- Aggregator:
  - A. verify $n$ signatures
  - B. aggregate $n$ ciphertexts
  - C. make 1 signature
- Control Center:
  - A. verify 1 signature
  - B. decrypt ciphertext
  - C. make decisions
- Decisions
**Problem statement**

- Two small wrinkles:
  - Complex **signature** and **verification** operations

### 2012 TPDS: EPPA

<table>
<thead>
<tr>
<th></th>
<th>2012 TPDS: EPPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign &amp; Ver Cost</td>
<td>((N + 1) \times T_m + (N + 1) \times T_p)</td>
</tr>
<tr>
<td>Total Cost</td>
<td>((2N + 5) \times T_m + T_e + (N + 9)T_p)</td>
</tr>
</tbody>
</table>

### Notations and Descriptions

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_m)</td>
<td>Multiplication operation</td>
</tr>
<tr>
<td>(T_e)</td>
<td>Exponentiation operation</td>
</tr>
<tr>
<td>(T_p)</td>
<td>Pairing operation</td>
</tr>
</tbody>
</table>

### 2014 TII: PEDA

<table>
<thead>
<tr>
<th></th>
<th>2014 TII: PEDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign &amp; Ver Cost</td>
<td>((N + 1) \times T_m + (2N + 1) \times T_e + (N + 1) \times T_p)</td>
</tr>
<tr>
<td>Total Cost</td>
<td>(3N \times T_m + (5N + 1) \times T_e + (N + 1) \times T_p)</td>
</tr>
</tbody>
</table>

- Aggregator is always **resource-constraint**
Edge Computing

- Edge computing architecture

- How to apply the new signature to traditional PPDA while ensure users data privacy?

- How to construct a new signature method to solve the computational problem in PPDA?

Lightweight PPDA
Our work

- LPDA: System model
  - Shifting the time-consuming operations to ES

<table>
<thead>
<tr>
<th>Entities</th>
<th>Trusted Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>Fully trusted</td>
</tr>
<tr>
<td>CC</td>
<td>Honest-but-curious</td>
</tr>
<tr>
<td>ES</td>
<td>Honest-but-curious</td>
</tr>
<tr>
<td>ETs</td>
<td></td>
</tr>
<tr>
<td>Adversary</td>
<td>Malicious</td>
</tr>
</tbody>
</table>
BLS signature scheme (BLS‘01: Asymmetric version)

- **KeyGen**: output \([g_1, pk = (g_1)^\alpha], sk \leftarrow \alpha\)
- **Sign** \((sk, m)\): output \(\sigma^{BLS} \leftarrow H(m)^\alpha\)
- **Verify** \((pk, m, \sigma^{BLS})\): accept if \(e(H(m), pk) = e(\sigma^{BLS}, g_1)\)

\[ e(H(m), pk) = e(H(m), (g_1)^\alpha) = e(H(m)^\alpha, g_1) = e(\sigma^{BLS}, g_1) \]

**Property**

- Signature aggregation: anyone can compress n signatures into one

\[ pk_1, m_1 \rightarrow \sigma_1 \]
\[ \vdots \]
\[ pk_n, m_n \rightarrow \sigma_n \]

\[ \text{Aggregate} \rightarrow \sigma^* \]

Verify \((pk, m, \sigma^*)\) = “accept”

Convinces verifier that:

User i signed the msg \(m_i\)
Offline signature:

- Calculate the DTCH function value: \( H_{ch_i} = g_1^{r_i} \cdot g_2^{s_i} \cdot g_3^{u_i} \)

  State information: \( St = (r_i, s_i, u_i) \)

- Let DTCH be the "msg": \( \sigma_i^{BLS} = (H_0(H_{ch_i}))^\alpha \)

- The verify phase is the same as BLS signature

Online signature:

- Chooses \( s'_i \) as a trapdoor random number

- Generate the online signature:

  \[
  \sigma_i^{on} = u'_i = (r_i - c_i) + (s_i - s'_i)y + u_izz^{-1}
  \]

- Verify: \( H_{ch}(r_i, s_i, u_i) = H_{ch}(c_i, s'_i, u'_i) \)
Aggregation Phase

- **Ciphertext aggregation: Pailler’97**
  - Paillier homomorphic encryption: \( c_i = g^{m_i} \cdot v_i^n \mod n^2 \)
  - Ciphertexts aggregation: \( c = \prod_{i=1}^{\omega} c_i \mod n^2 = g^m \cdot \prod_{i=1}^{\omega} v_i^n \mod n^2 \)
  - Decryption: \( m = \sum_{i=1}^{\omega} m_i = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n \)

- **Offline signature aggregation**
  - Signature aggregation: \( \prod_{i=1}^{\omega} \sigma_i^{BLS} = \prod_{i=1}^{\omega} (H_0(H_{ch_i}))^\alpha \)
  - Verification:
    \[
    \prod_{i=1}^{\omega} e(H_0(H_{ch_i}), pk) = \prod_{i=1}^{\omega} e(H_0(H_{ch_i}), g_1^\alpha) = \prod_{i=1}^{\omega} e(H_0(H_{ch_i})^\alpha, g_1) = \prod_{i=1}^{\omega} e(\sigma_i^{BLS}, g_1) = e(\prod_{i=1}^{\omega} \sigma_i^{BLS}, g_1)
    \]
Applying OOS to PPDA scheme

**Offline Phase**

- Registration
  - Authentication
  - Before the \textit{msg} is given
    - State information
    - DTCH function value
    - BLS signature
  - Offline signature

**Online Phase**

- Online signature
  - Data aggregation
  - After the \textit{msg} is given
    - Random number
    - DTCH function value
    - State information
  - Online signature verification

Data encryption
Computational complexity comparison

Settings: Linux environment, PBC library.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
<th>Time Cost (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{E_1}$</td>
<td>Exponentiation Operation in $\mathbb{Z}_n^2$</td>
<td>1.58</td>
</tr>
<tr>
<td>$T_{E_2}$</td>
<td>Exponentiation Operation in $\mathbb{G}$</td>
<td>1.62</td>
</tr>
<tr>
<td>$T_M$</td>
<td>Multiplication Operation in $\mathbb{G}$</td>
<td>0.06</td>
</tr>
<tr>
<td>$T_P$</td>
<td>Pairing Operation</td>
<td>17.62</td>
</tr>
</tbody>
</table>

(a) Signature and Verification Cost Comparison

(b) Overall Computational Cost Comparison
Performance: Communication

- Communication overhead comparison

Our scheme is more efficient in both ET-to-ES and ES-to-CC communication overheads!
We proposed an online/offline signature and verification scheme, OOS, for edge computing which is proved existentially unforgeable under chosen message attacks.

We further apply the OOS scheme to the traditional PPDA scheme, and realize the lightweight privacy-preserving data aggregation with edge computing.

We conduct the numerical evaluation of the proposed LPDA-EC scheme, the results indicate that the time of signature and verification for edge terminals are small and constant.
Thanks a lot!

Questions?