南京航空航天大学

NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

TEMPLE UNIVERSITY®

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

Jiale Zhang, Yanchao Zhao, Jie Wu & Bing Chen

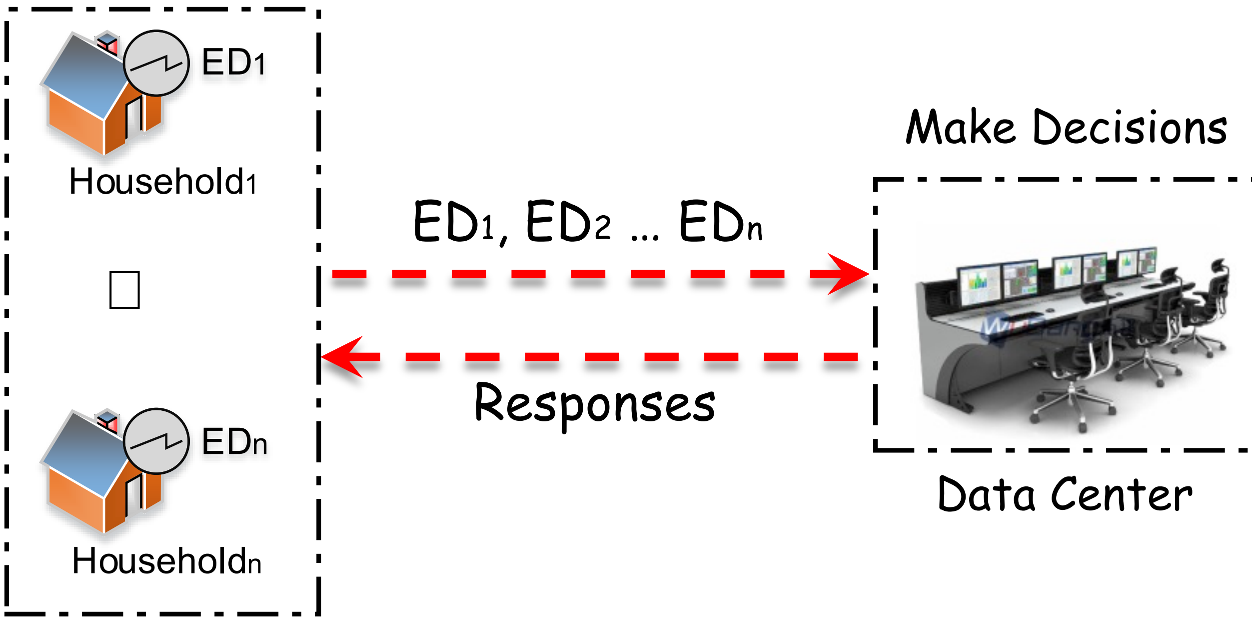
@ Nanjing University of Aeronautics and Astronautics & Temple University

IEEE MASS 2018

Chengdu, China

Background

☐ Simple application scenario: smart grid



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

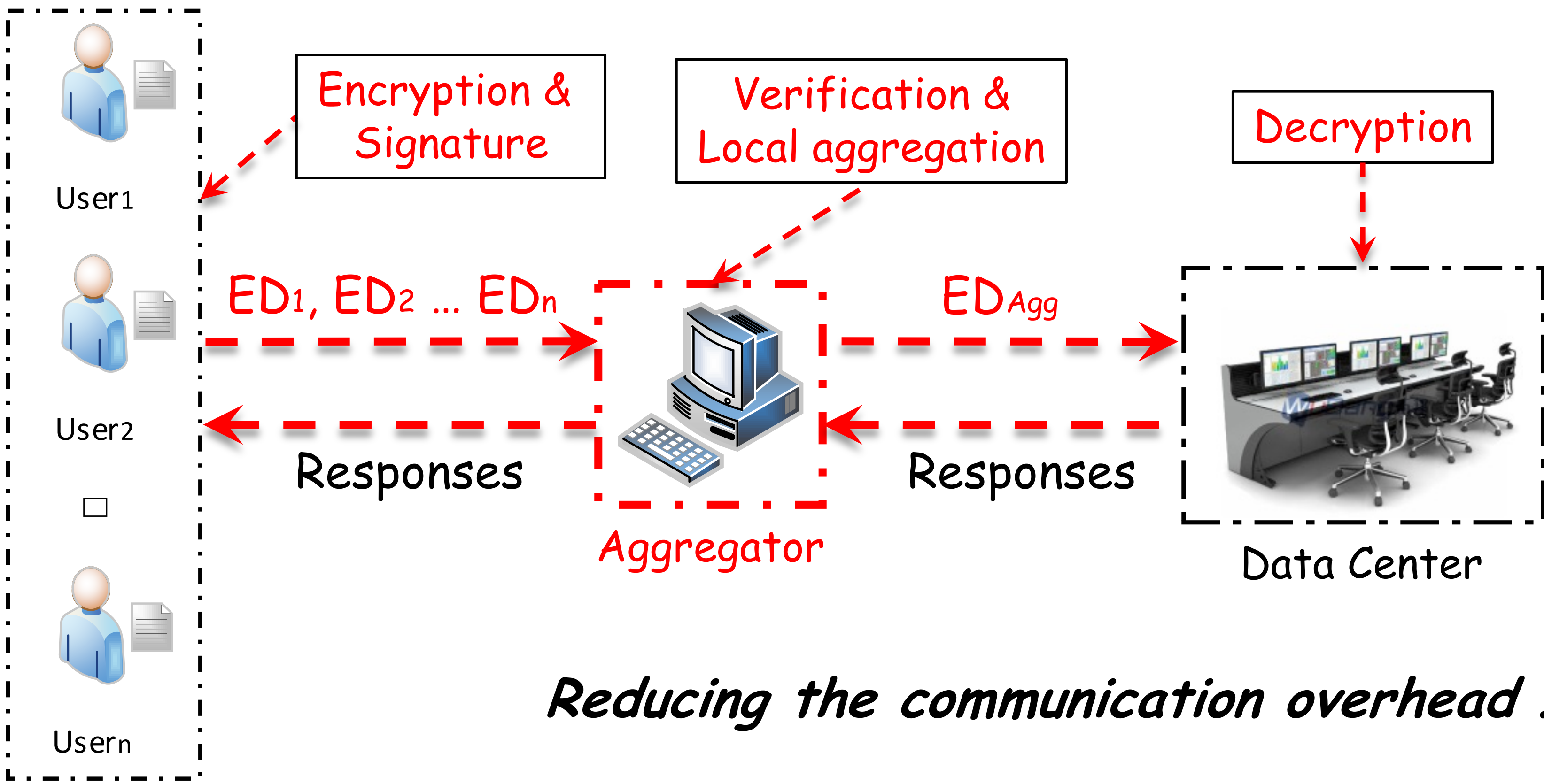
Background

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



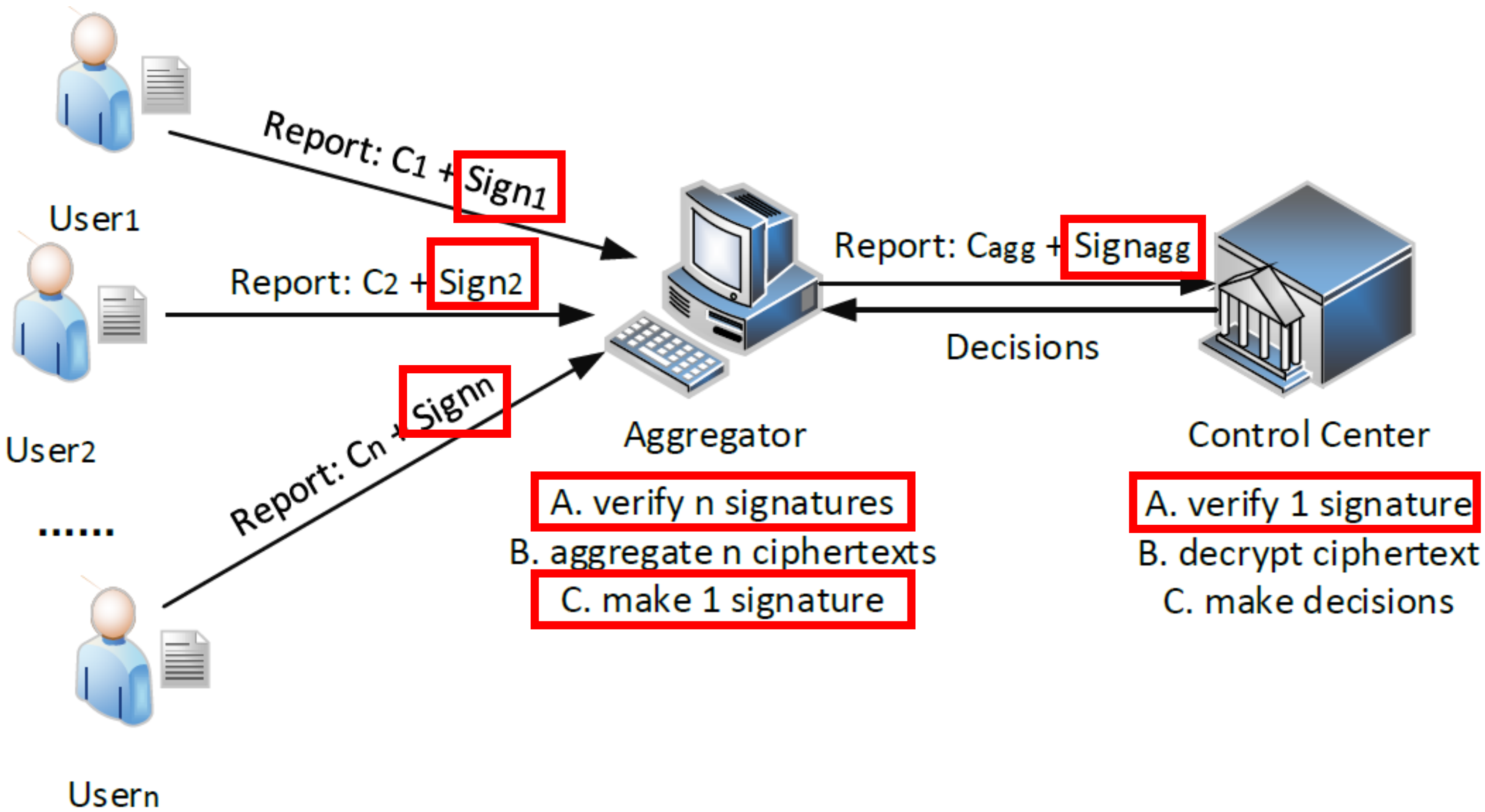
Reducing the communication overhead !

Problem statement

- 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



Problem statement

- Two small wrinkles:
  - Complex signature and verification operations

2012 TPDS: EPPA	
Sign & Ver Cost	$(N+1) * T_m + (N+1) * T_p$
Total Cost	$(2N+5) * T_m + T_e + (N+9)T_p$

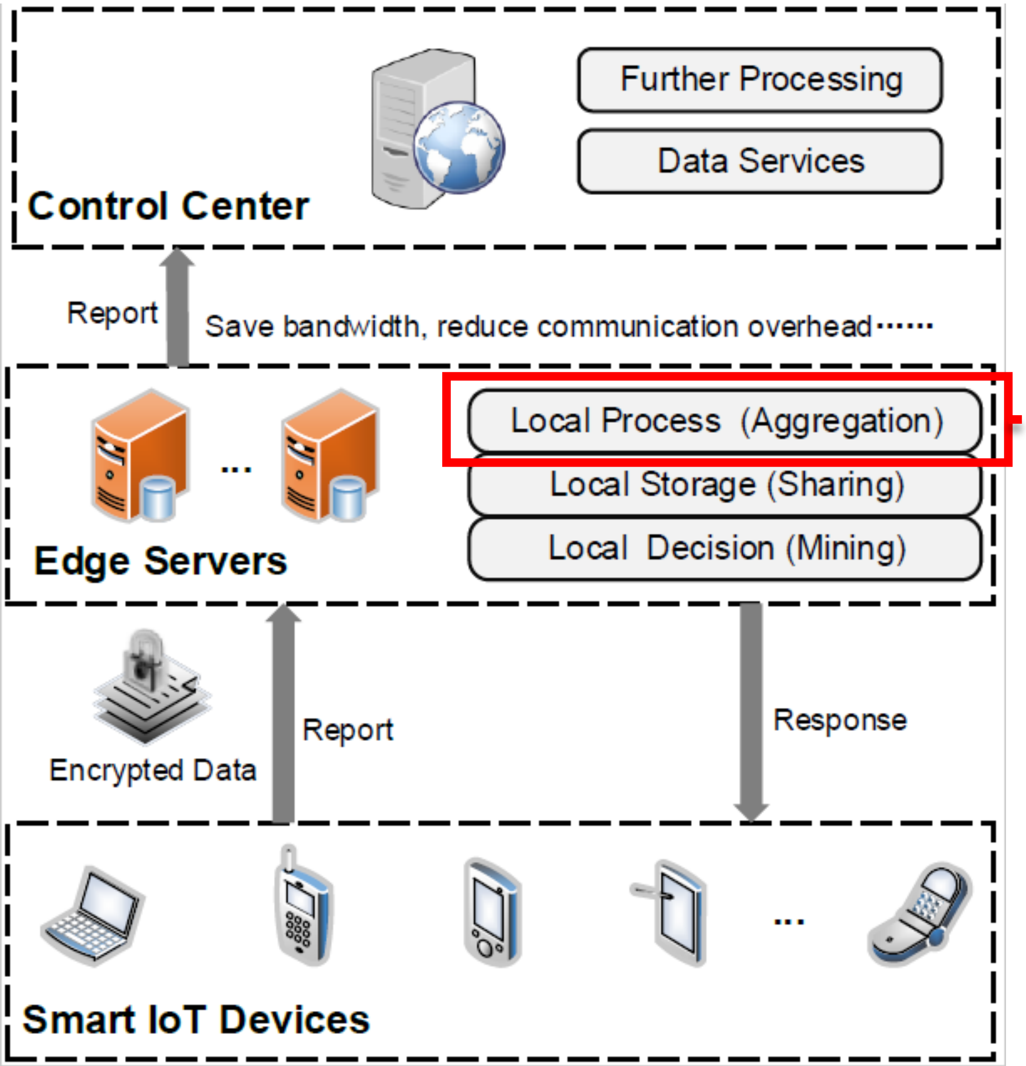
Notations	Descriptions
T_m	Multiplication operation
T_e	Exponentiation operation
T_p	Pairing operation

2014 TII: PEDA	
Sign & Ver Cost	$(N+1) * T_m + (2N+1) * T_e + (N+1) * T_p$
Total Cost	$3N * T_m + (5N+1) * T_e + (N+1) * T_p$

  - Aggregator is always resource-constraint

Edge Computing

□ Edge computing architecture



Lightweight PPDA

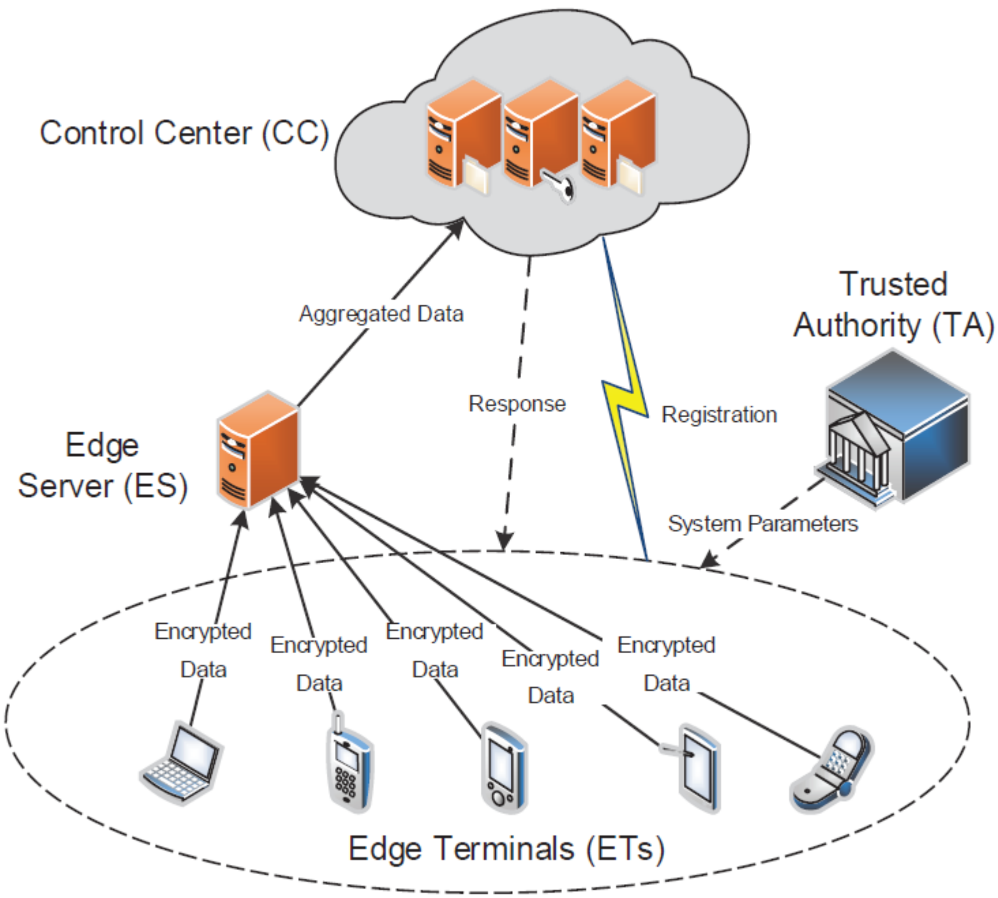
How to construct a new signature method to solve the computational problem in PPDA?

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

Our work

LPDA: System model

- Shifting the time-consuming operations to ES



Entities	Trusted Model
TA	Fully trusted
CC	Honest-but-curious
ES	Honest-but-curious
ETs	
Adversary	Malicious

OOS: Online/offline signature

BLS signature scheme (BLS'01: Asymmetric version)

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

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

Property

- Signature aggregation: <u>anyone</u> can compress n signatures into <u>one</u>

$pk_1, m_1 \rightarrow \sigma_1$

$\vdots$

$pk_n, m_n \rightarrow \sigma_n$

Aggregate $\rightarrow \sigma^*$

Verify (pk, m, σ^*) = "accept"

Convinces verifier that:

User i signed the msg m_i

OOS Construction

Offline signature:

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

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

- Let DTCH be the "msg": $\sigma_i^{BLS} = (\boxed{H_0(H_{ch_i})})^{\alpha}$
- The verify phase is the same as BLS signature

Online signature:

- Chooses $\boxed{s_i'}$ as a trapdoor random number
- Generate the online signature:

$$\sigma_i^{on} = u_i' = ((r_i - \boxed{c_i}) + (s_i - s_i')y + u_i z)z^{-1}$$

- Verify: $H_{ch}(r_i, \boxed{s_i}, u_i) = H_{ch}(c_i, \boxed{s_i'}, u_i')$

Aggregation Phase

Ciphertext aggregation: Pailler'97

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

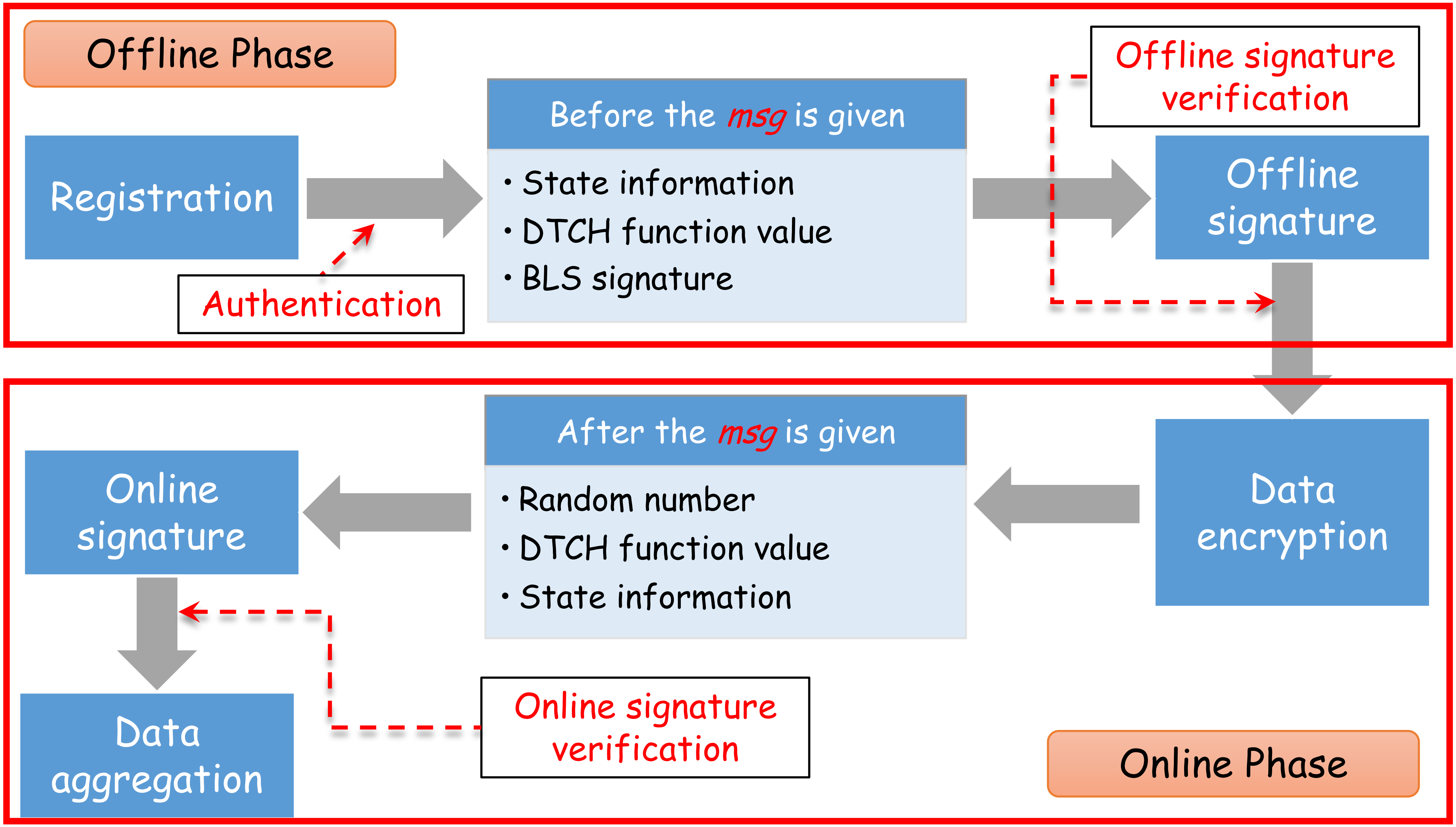
Offline signature aggregation

- Signature aggregation: $\prod_{i=1}^{\omega} \sigma_i^{BLS} = \prod_{i=1}^{\omega} \left(H_0(H_{ch_i})\right)^{\alpha}$
- Verification: $\prod_{i=1}^{\omega} e(H_0(H_{ch_i}), pk) = \boxed{\prod_{i=1}^{\omega} e(H_0(H_{ch_i}), g_1^{\boxed{\alpha}})} =$

$$\boxed{\prod_{i=1}^{\omega} e(H_0(H_{ch_i})^{\boxed{\alpha}}, g_1)} = \prod_{i=1}^{\omega} e(\sigma_i^{BLS}, g_1) = e(\boxed{\prod_{i=1}^{\omega} \sigma_i^{BLS}}, g_1)$$

LPDA-EC Construction

- Applying OOS to PPDA scheme

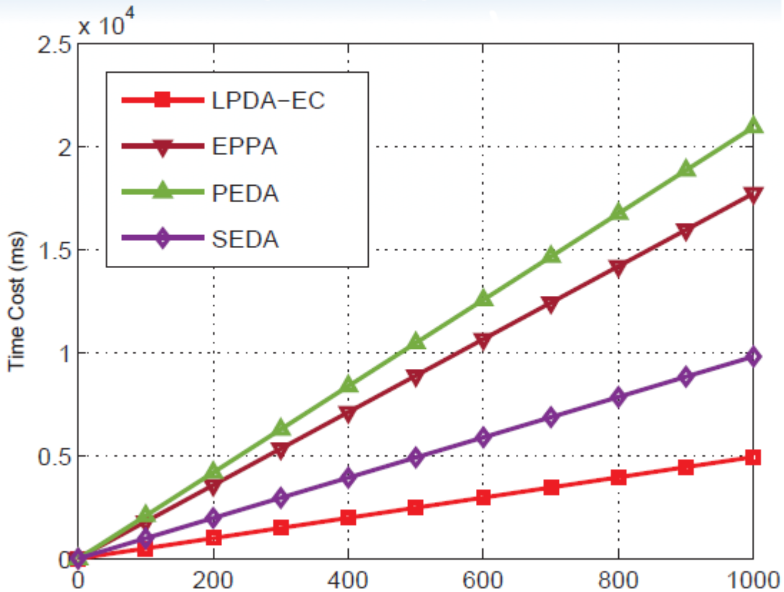


Performance: Computational

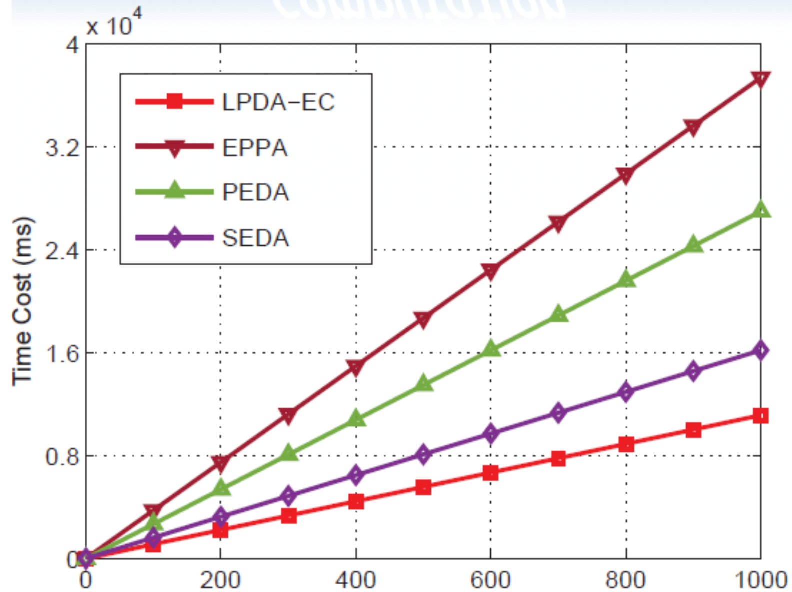
- Computational complexity comparison

Settings: Linux environment, PBC library.

Notations	Description	Time Cost (ms)
T_{E_1}	Exponentiation Operation in $\mathbb{Z}_{n^2}$	1.58
T_{E_2}	Exponentiation Operation in $\mathbb{G}$	1.62
T_M	Multiplication Operation in $\mathbb{G}$	0.06
T_P	Pairing Operation	17.62



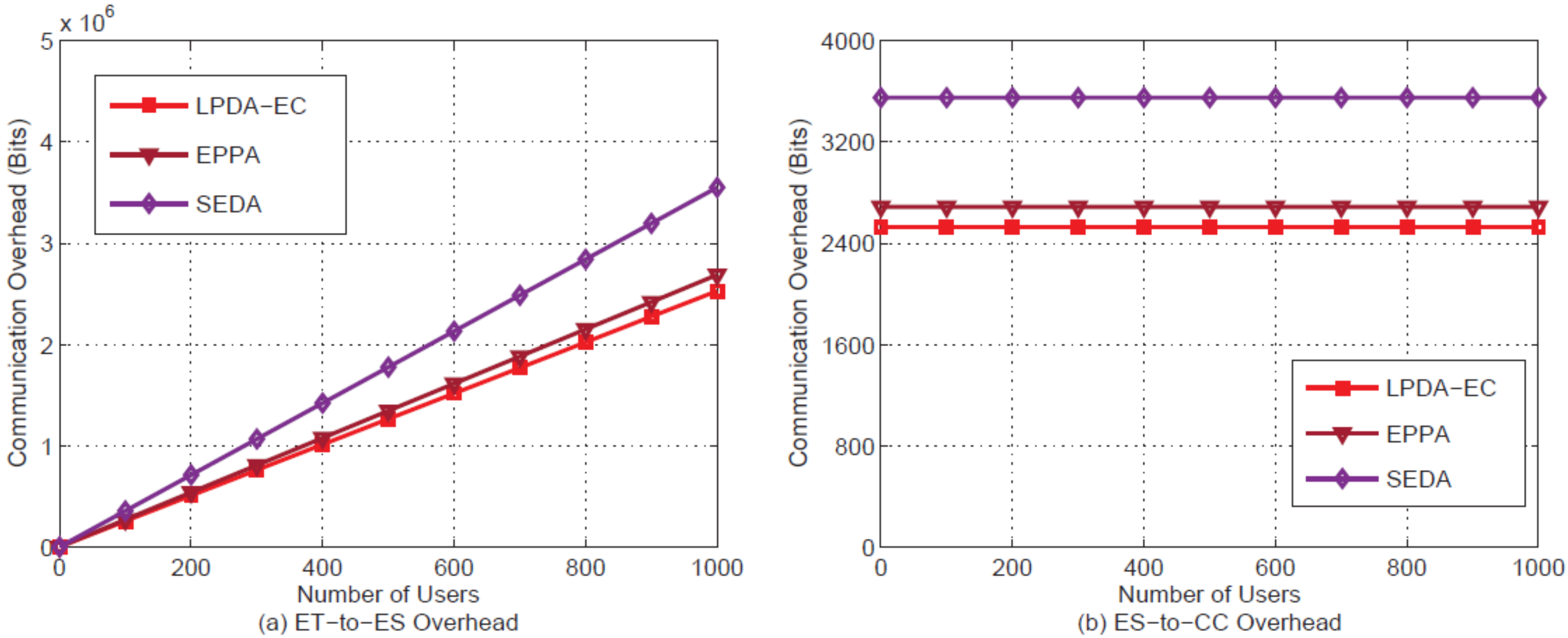
(a) Signature and Verification Cost Comparison



(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!

Conclusion

- 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?