PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks

IEEE INFOCOM, 2007

Presented by Chia-Yi Lien
January 3, 2008

Outline

- Introduction
- Model and Background
- Private Data Aggregation Protocols
- Evaluation
- Conclusion

Introduction (1/2)

- Providing efficient data aggregation while preserving data privacy is a challenging problem in wireless sensor networks research.
- The goal of our work is to bridge the gap between collaborative data collection by wireless sensor networks and data privacy.

Introduction (2/2)

- To the best of our knowledge, this paper is among the first on privacy-preserving data aggregation in wireless sensor networks.
- In this paper, we focus on additive aggregation functions, that is, $f(t) = \sum_{i=1}^{N} d_i(t)$
  - $d_i(t)$ is the individual sensor reading at time t for node i

Model and Background

- Desirable characteristics of a private data aggregation scheme
  - Privacy
    - Each node's data should be only known to itself
  - Efficiency
    - A good private data aggregation scheme should keep the overhead which is introduced to protect privacy as small as possible
  - Accuracy

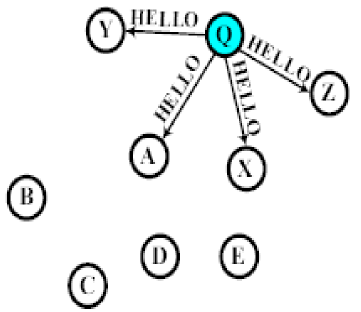
Private Data Aggregation Protocols

- Cluster-based Private Data Aggregation (CPDA)
  - Advantage: less communication overhead
- Slice-Mix-AggRegaTe (SMART)
  - Advantage: less computation overhead
- When there is no packet loss, in both CPDA and SMART, the sensor network can obtain a precise aggregation result while guaranteeing that no private sensor reading is released to other sensors.

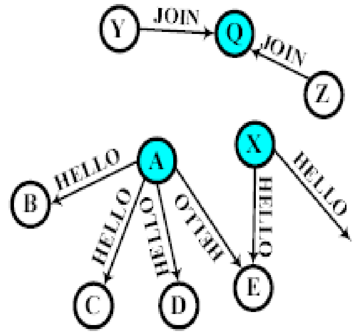
CPDA

- it guarantees that no individual node knows the data values of other nodes.
- CPDA consists of three phases
  - Cluster formation
    - A sensor elects itself as a cluster leader with a probability p_c
  - Calculation within clusters
  - Cluster data aggregation

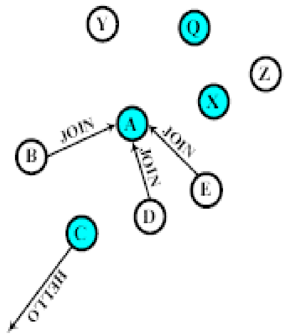
Cluster formation



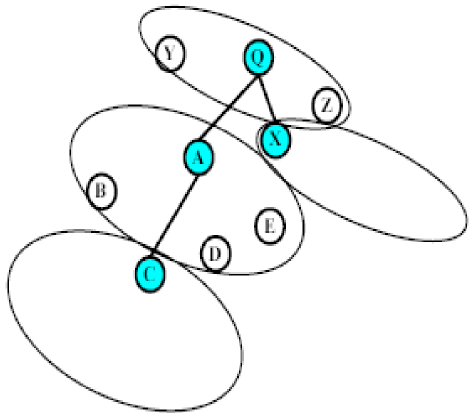
(a) Query Server Q triggers a query by *HELLO* message. A recipient of *HELLO* message elects itself as a cluster leader randomly.



(b) *A* and *X* become cluster leader, so they broadcast the *HELLO* message to their neighbors.



(c) Node *E* receives multiple *HELLO* messages, then *E* randomly selects one to join.



(d) Several clusters have been constructed and the aggregation tree of cluster leaders is formed

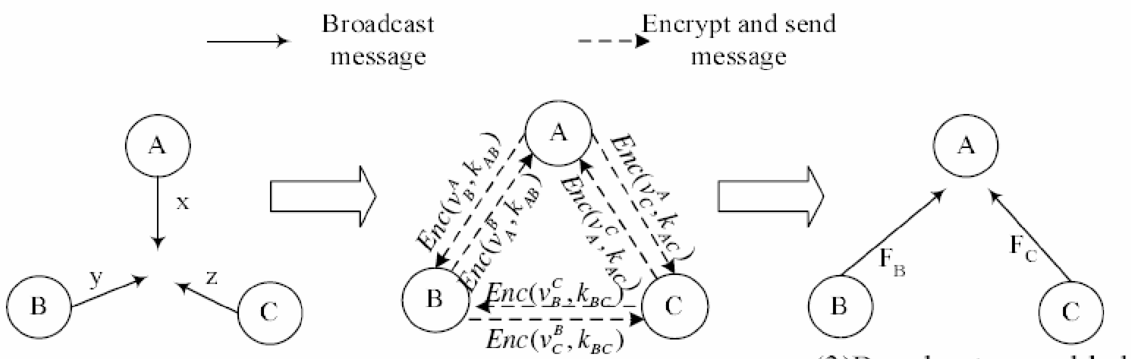
Calculation within clusters

Node A:
$$v_A^A = a + r_1^A x + r_2^A x^2,$$
$$v_B^A = a + r_1^A y + r_2^A y^2,$$
$$v_C^A = a + r_1^A z + r_2^A z^2,$$

$$NodeB: \begin{aligned} v_A^B &= b + r_1^B x + r_2^B x^2, \\ v_B^B &= b + r_1^B y + r_2^B y^2, \\ v_C^B &= b + r_1^B z + r_2^B z^2. \end{aligned}$$

$$NodeC: \begin{aligned} v_A^C &= c + r_1^C x + r_2^C x^2, \\ v_B^C &= c + r_1^C y + r_2^C y^2, \\ v_C^C &= c + r_1^C z + r_2^C z^2. \end{aligned}$$

$$F_A = v_A^A + v_A^B + v_A^C = (a + b + c) + r_1 x + r_2 x^2,$$
$$F_B = v_B^A + v_B^B + v_B^C = (a + b + c) + r_1 y + r_2 y^2,$$
$$F_C = v_C^A + v_C^B + v_C^C = (a + b + c) + r_1 z + r_2 z^2.$$



(1)Broadcast seeds (2)Encrypt and send customized values (3)Broadcast assembled information F_A, F_B, F_C

Aggregate value

$$\begin{bmatrix} a+b+c \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \end{bmatrix}^{-1} [F_A, F_B, F_C]^T$$

Cluster data aggregation

- Each cluster leader routes the derived sum within the cluster back towards the query server through a TAG routing tree rooted at the server

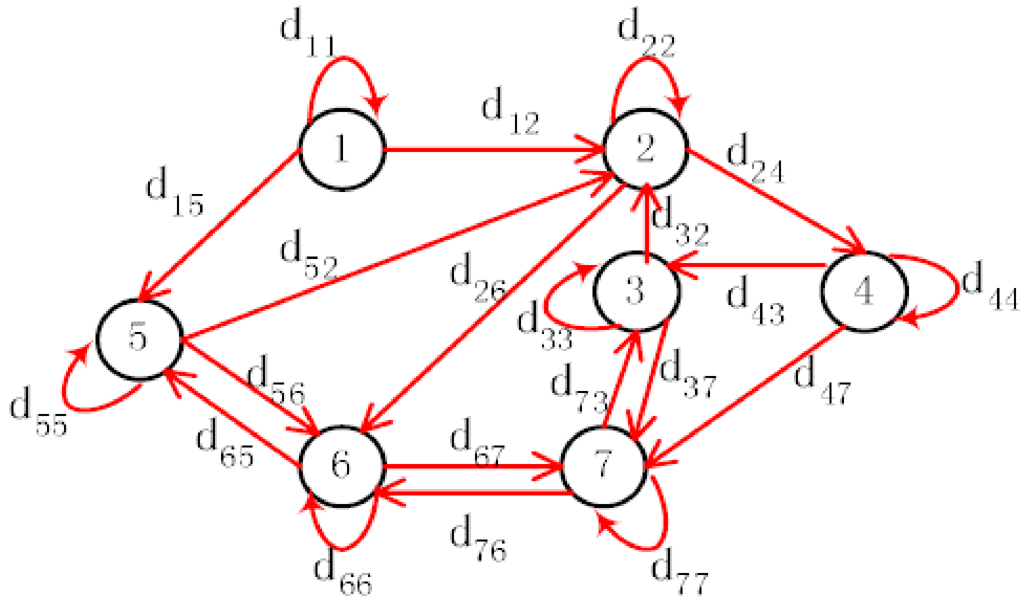
SMART (1/2)

- each node hides its private data by slicing it into pieces and sending encrypted data slices to different aggregators.
- Then the aggregators collect and forward data to a query server (sink).
- When the server receives the aggregated data, it calculates the final aggregation result.

SMART (2/2)

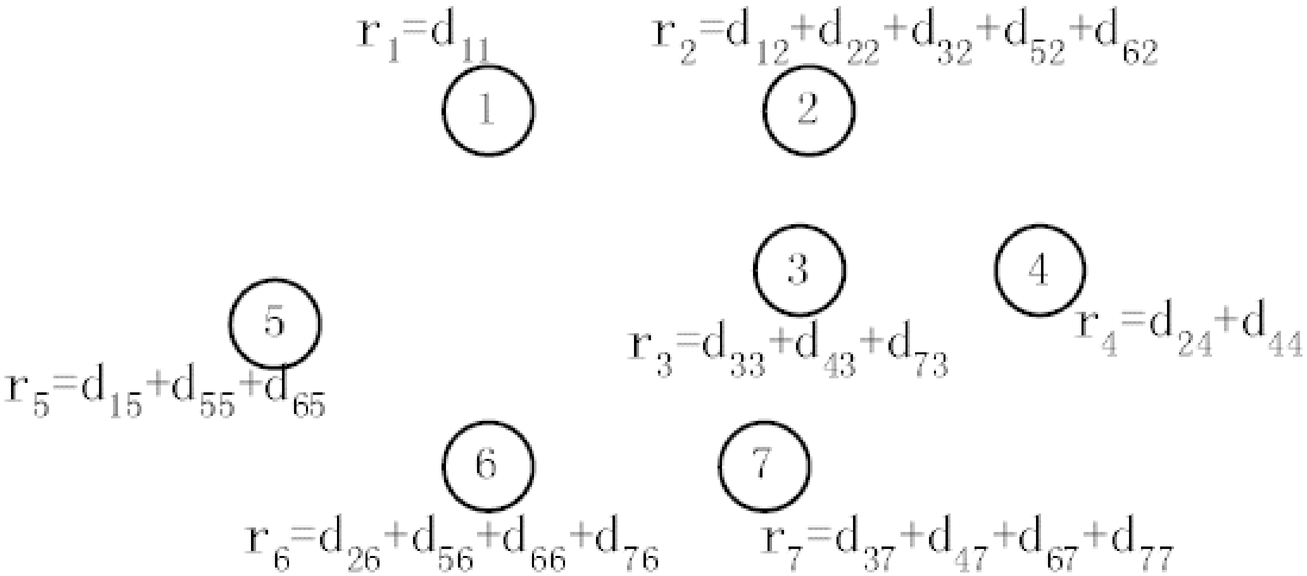
- Three Steps: slicing, mixing, aggregation
- Slicing
  - Each node randomly selects a set of nodes (J=|Si|) within h hops
  - One of the J pieces is kept at node i itself. The remaining $J-1$ pieces are encrypted and sent to nodes in the randomly selected set S_i

SMART - Slicing



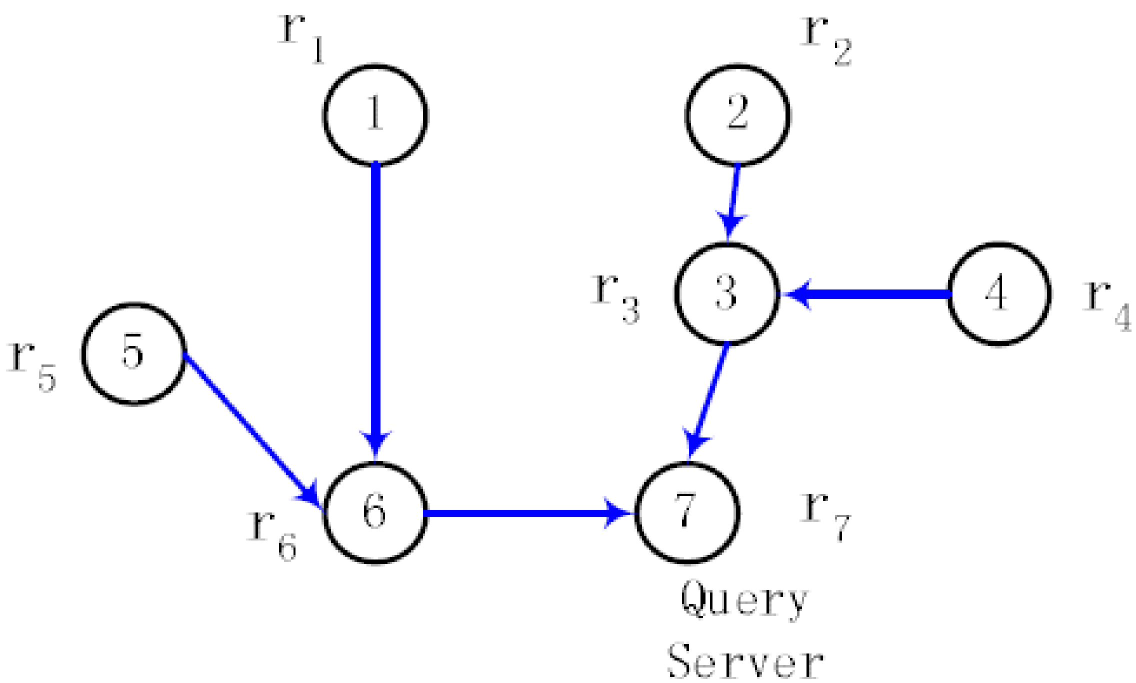
(a) Slicing ($J = 3, h = 1$):$d_{ij}(i \neq j)$ is encrypted and transmitted from node i to j, where $j \notin S_i$. d_{ii} is the data piece kept at node i.

SMART - Mixing



(b) Mixing: Each node i decrypts all data pieces received and sums them up including the one kept at itself (d_{ii}) as r_i.

SMART – Aggregation

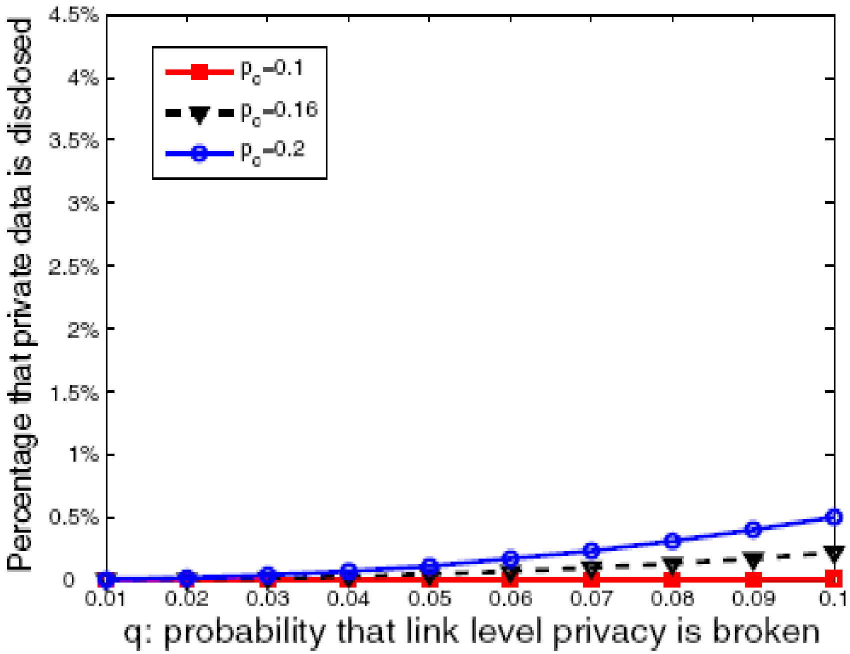


(c) Aggregation (No encryption is needed)

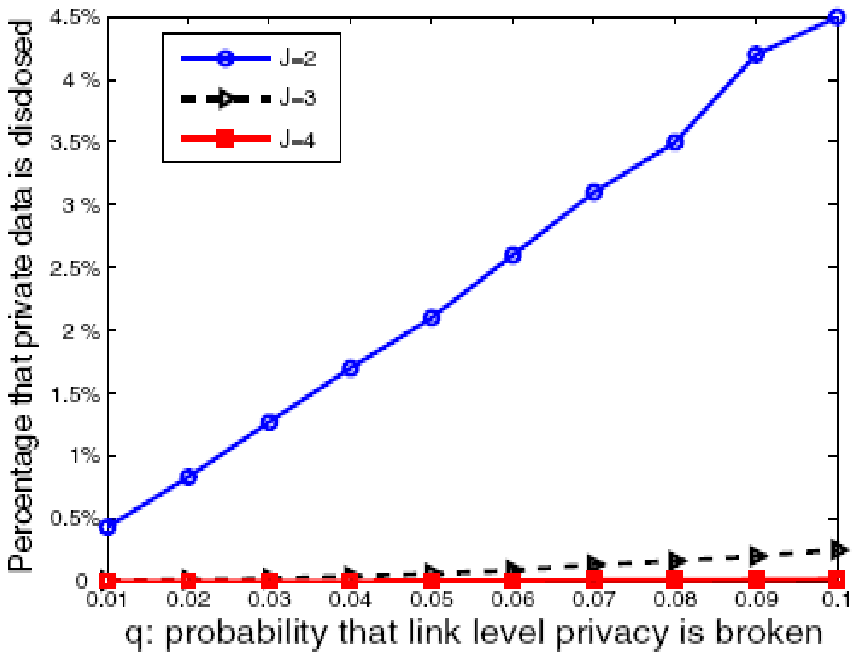
Evaluation

- Compare with a commonly used data aggregation scheme – TAG (Tiny AGgregation), where no data privacy protection is provided

Privacy-preservation Efficacy



(a) CPDA

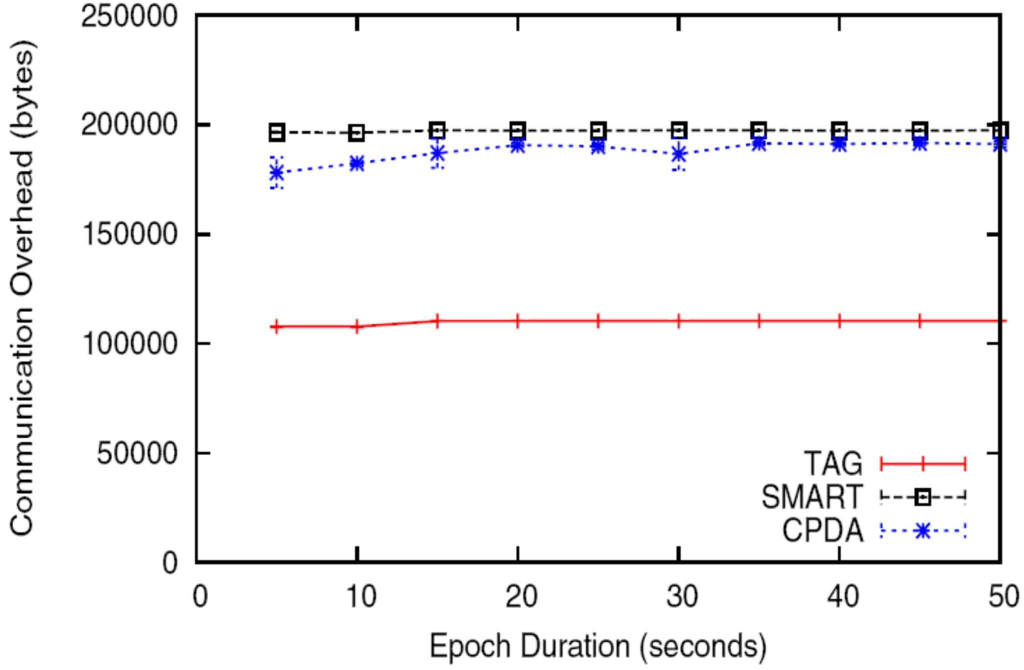


(b) SMART

Communication Overhead (1/3)

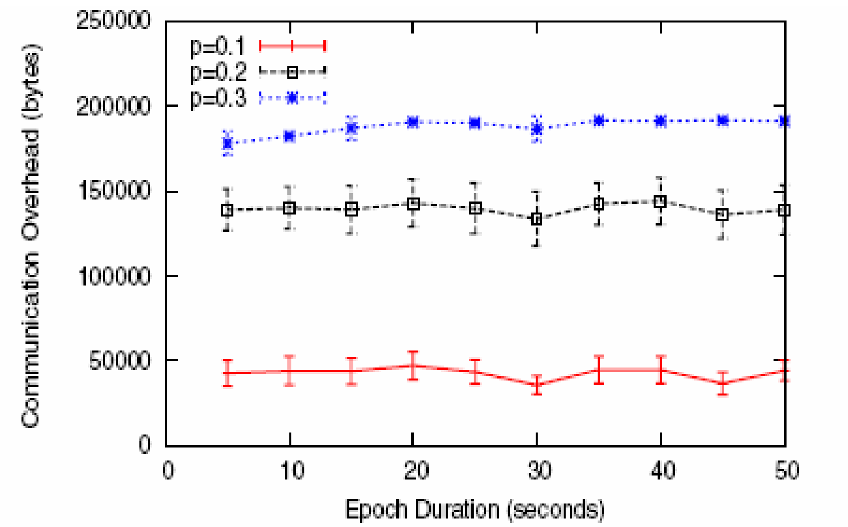
- Epoch duration is the amount of time for the data aggregation procedure to finish
- In *TAG*, each node needs to send 2 messages for data aggregation: one *Hello* message to form an aggregation tree, and one message for data aggregation.
- $3+p_c$ is the average number of messages sent by a node in *CPDA*. Thus, the overhead in *CPDA* is less than twice as that in *TAG*.
- *SMART*, with $J = 3$, needs to exchange 2 messages during the slicing step and 2 messages for data aggregation. Therefore, the overhead of *SMART* is double that of *TAG*.

Communication Overhead (2/3)

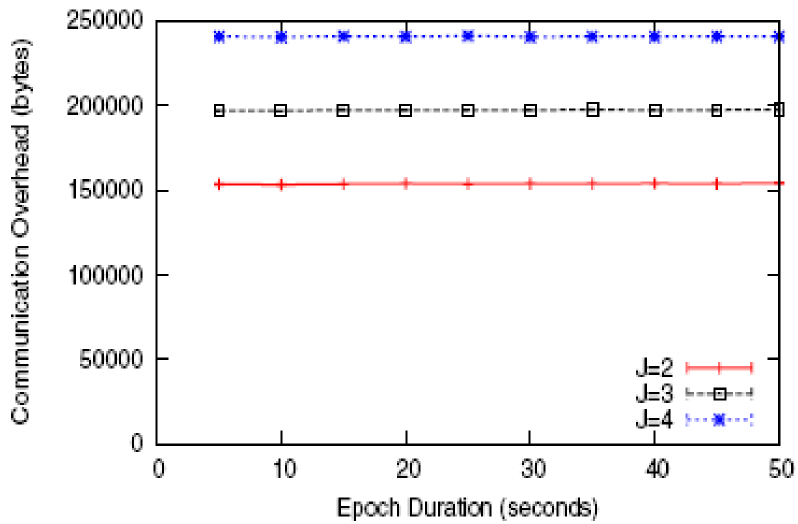


(a) Comparison of TAG, CPDA ($p_c = 0.3$) and SMART (J=3).

Communication Overhead (3/3)

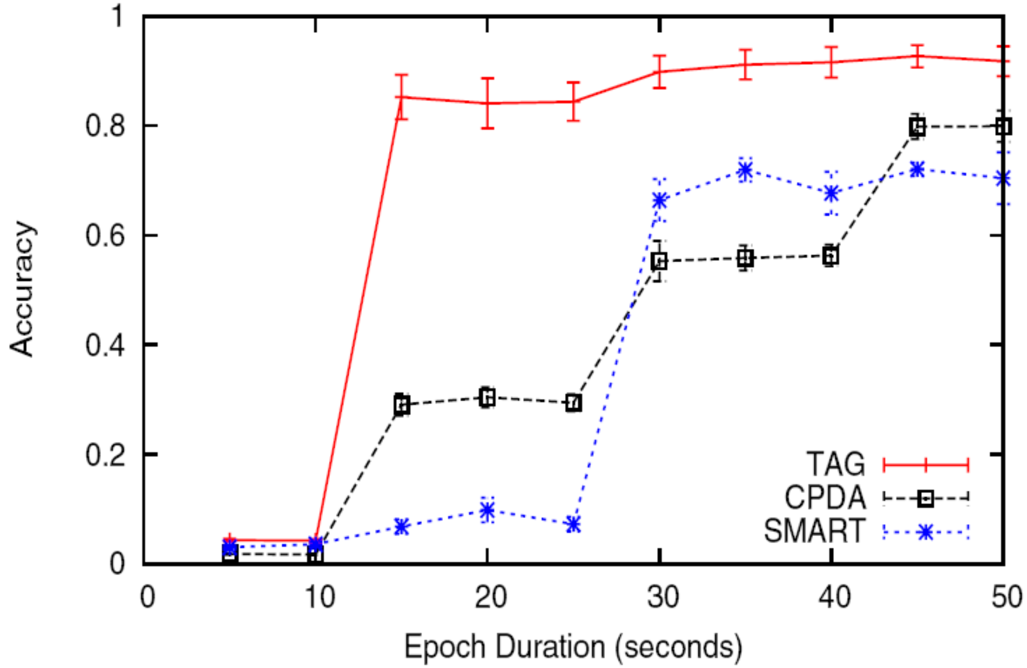


(b) Communication overhead of CPDA with respect to p_c.



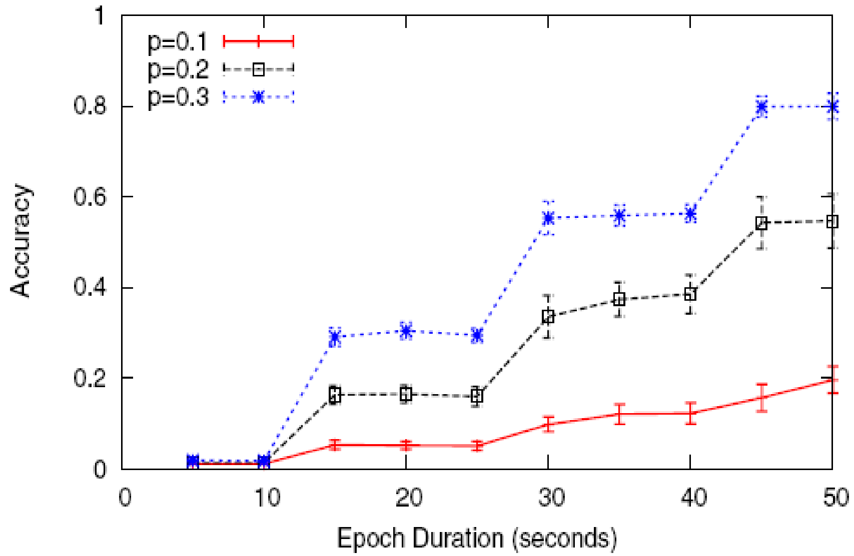
(c) Communication overhead of SMART with respect to J.

Accuracy (1/2)

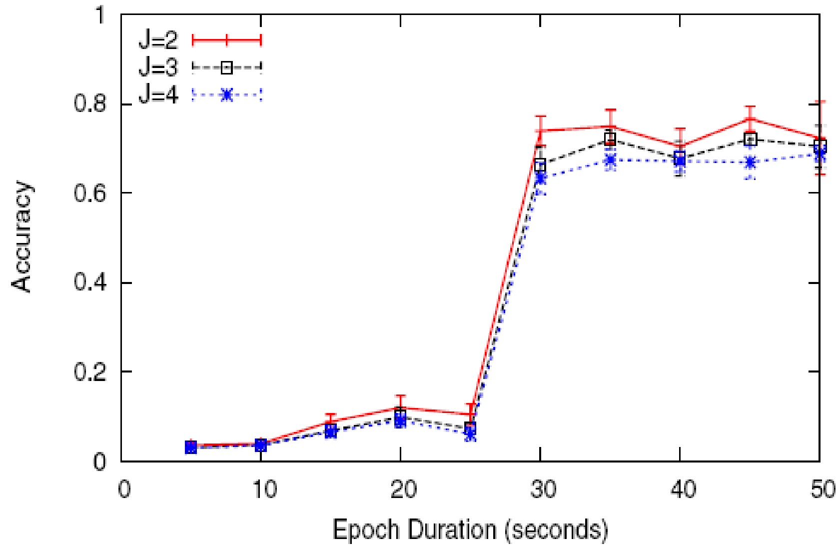


(a) Accuracy comparison of TAG, CPDA ($p_c = 0.3$) and SMART (J=3).

Accuracy (2/2)



(b) Accuracy of CPDA with respect to p_c.



(c) Accuracy of SMART with respect to J.

Conclusion

- CPDA and SMART use data-hiding techniques and encrypted communication to protect data privacy
- We propose two private-preserving data aggregation schemes – *CPDA*, and *SMART* – focusing on additive data aggregation functions.
- PERFORMANCE COMPARISON OF CPDA AND SMART

	CPDA	SMART
Privacy preservation efficacy	Excellent	Excellent ($J \geq 3$)
Communication overhead	Fair	Large
Aggregation accuracy	Good (but sensitive to p_c)	Good (not sensitive to J)
Computational overhead	Fair	Small