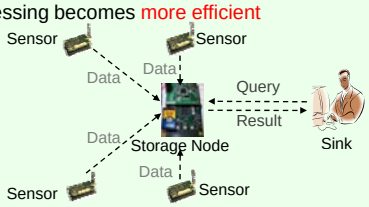
ACM CCS 2009

Privacy and Integrity Preserving Range Queries in Sensor Networks

Fei Chen and Alex X. Liu
{feichen, alexliu}@cse.msu.edu

Motivation

- The architecture of two-tired sensor networks has been widely adopted
  - Power saving and Memory saving for sensors
  - Query processing becomes more efficient



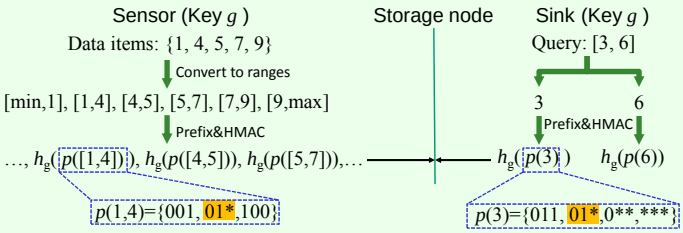
- Storage nodes are attractive to be attacked.
  - Sensitive data collected by sensors are stored in them
- How can we preserve the privacy and integrity of query result if a storage node is compromised?
  - Storage nodes cannot gain information from data and queries
  - Storage nodes can perform query processing
  - The sink can detect whether a query result from a storage node
    - Includes forged data items
    - Excludes any data items that satisfy the query

Contributions

- We propose oblivious membership verification for preserving privacy,
- We propose neighborhood chaining for preserving integrity,
- We propose an optimization technique using Bloom filters to significantly reduce the communication cost between sensors and storage nodes.
- We propose a solution for event-driven sensor networks.
- Our solution is more secure and more efficient than the state-of-the-art.

Privacy Preserving for 1-dimensional Data

- To preserve the privacy of sensor collected data
  - Encrypt each data item individually, e.g. $(1)_{k_i}$,$(4)_{k_i}$,$(5)_{k_i}$,$(7)_{k_i}$,$(9)_{k_i}$.
- How does a storage node process a query over encrypted data?
  - Use prefix membership verification technique.



If two sets have a common element, $3 \in [1,4]$. Similarly, $6 \in [5,7]$. Therefore, the query result is $(3)_{k_i}$,$(5)_{k_i}$

Integrity Preserving for 1-dimensional Data

- We propose neighborhood chaining technique
  - Encrypt the data item with its left neighbor

$(\min|1)_{k_i}$ $(1|4)_{k_i}$ $(4|5)_{k_i}$ $(5|7)_{k_i}$ $(7|9)_{k_i}$ $(9|\max)_{k_i}$

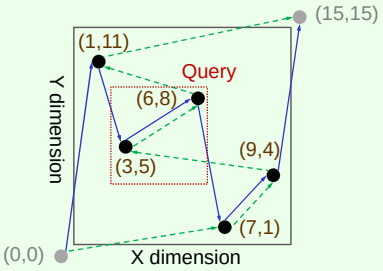
Range [3, 6]

Query result Verification object

- The storage node cannot exclude
  - Any item in the middle, e.g. $(4|5)_{k_i}$
  - The first item $(1|4)_{k_i}$
  - The last item $(5|7)_{k_i}$

Privacy and integrity for Multi-dimensional Data

- To preserve privacy, we apply our 1-dimensional privacy preserving techniques to each dimension of multi-dimensional data.
- To preserve integrity, we build a multi-dimensional neighborhood chain.



The multi-dimensional neighborhood chain of the above example is $(0|1, 9|11)_{k_i}$, $(1|3, 4|5)_{k_i}$, $(3|6, 5|8)_{k_i}$, $(6|7, 0|1)_{k_i}$, $(7|9, 1|4)_{k_i}$, $(9|15, 11|15)_{k_i}$,

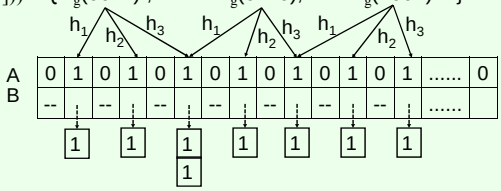
Optimization

- Communication cost between sensors and storage nodes is significant.
  - The number of prefixes in $p([d_j,d_{j+1}])$ is at most $2w-2$, w is the bit length of d_j
  - Each $h_g(\text{prefix})$ is 128-bit (if use HMAC-MD5) or 160-bit (if use HMAC-SHA1)
- A sensor uses a Bloom filter to represent hashes, i.e.

$h_g(p([\min,1]), h_g(p([1,4])), h_g(p([4,5])), h_g(p([5,7]), h_g(p([7,9]), h_g(p([9,\max])$

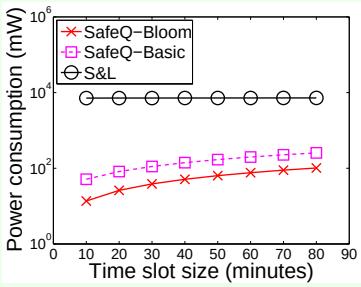
Index: 0 1 2 3 4 5

$h_g(p([1,4])) = \{ h_g(0011), h_g(0110), h_g(1001) \}$

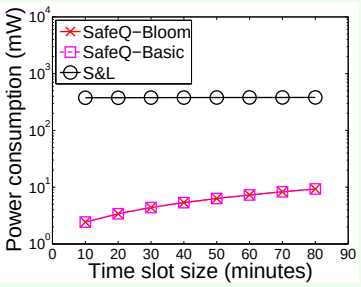


Experimental Results

- We conducted experiments on both S&L (the state-of-art) and our solution
  - In terms of power consumption, for 3-dimensional data, our scheme is 184.9 times less power for sensors and 76.8 times less power for storage nodes



Power consumption for storage nodes



Power consumption for storage nodes

  - In terms of space consumption, for 3-dimensional data, our scheme is 182.4 times less space for storage node