An IoT Data Communication Framework for Authenticity and Integrity

Xin Li*, Huazhe Wang*, Ye Yu†, Chen Qian*

*University of California Santa Cruz
†University of Kentucky
IoT is ubiquitous
IoT trends

By 2020

4 Billion
Connected People

4 Trillion
Revenue

25+ Billion
Device

50 Trillion GB
Traffic
IoT Data Applications: data consumers

Analytics
• Environment monitoring
• Traffic estimation
• Business decision making

Prediction
• Whether forecast
• Electricity load forecast

Real-time control
• Autonomous car
• Manufacturing
• Smart lighting
IoT device hardware platforms

Problem:
Limited computation and storage capacity
IoT Communication Framework

Data application
Requirements and challenges

• Data sampling
  ❖ Due to bandwidth, storage quota limits
  ❖ Requirement: uniformity
Requirements and challenges

• Partial Data Retrieval
  - Different granularity requirements
  - Requirement: partial data retrieval, uniformity

Corse-grained traffic estimation
Requirements and challenges

• Partial Data Retrieval
  - Different granularity requirements
  - Requirement: partial data retrieval, uniformity

License plate recognition for toll way billing
Security threat

The Dirty, Little Secret of the Data Center — Data Corruption White

Computer hackers take to the cloud

Online data storage services often — and unknowingly — host malicious software

Wrong Decisions!
Security threat

Security mode: Only end entities are trust-worthy

Data application
Security threat

Scope: Authentication and integrity. Privacy is orthogonal.
Digital signature preliminary

\[ H(\text{...}) \]

- Alice
- Bob

Alice’s **private** key

Alice’s **public** key

Expensive & Slow
Digital signature preliminary

Alice  

Bob

Alice’s **private** key

Alice’s **public** key
Digital signature preliminary

Alice

Alice’s private key

Bob

Alice’s public key

Expensive & Slow
Digital signature scheme: sign-each

Problem: inefficient and no uniformity guarantee

Power hungry

Slow
Digital signature scheme: concatenate

Problem: partial data retrieval not supported
Digital signature scheme: Merkle tree

\[ D_{12} = H(D_1 || D_2) \]
Digital signature scheme: Merkle tree

Verifying

D14

D12

D34

H(M2) = D2

Single packet verifiable

No uniformity guarantee
## Signature scheme comparison

<table>
<thead>
<tr>
<th>Signature Scheme</th>
<th>Computation Efficiency</th>
<th>Partial Data Retrieval</th>
<th>Uniformity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign each</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Concatenate</td>
<td>✓</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Merkle tree</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>GSC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Geometric star chaining

- Intuition: any fraction number can be represented or approximated by a few bits

\[ 5/8 = (0.101)_2 \]
Geometric star chaining

\[ \frac{5}{8} = (0.101)_2 \]
Geometric star chaining

\[ D_1 = H(m||D_1) \]

Constant!

Dynamically updated
Budget limit

Limited storage quota

Limited bandwidth
Budged-based distributed stream sampling

Each epoch: budget = 8

Store and sample ->
Significant Space overhead
Min-wise sampling

Communication cost is reduced for the smallest changes.

- 0.391
- 0.908
- 0.291
- 0.555
- 0.619
- 0.273

Not Sent
Budged-based distributed stream sampling

- Coordinator is not trust-worthy
- Sampling is not compatible with GSC

Please check out the paper.
Sampling Protocol Design

Sensing device

Algorithm 1: SP at sensing device $k$ in round $j$

1. foreach event $e$ do
2. \hspace{1em} $i \leftarrow \arg \min_{x \in \mathbb{N}} \{h(e) \geq 2^{-x-1}\}$;
3. \hspace{1em} $l^k_i \leftarrow l^k_i + 1$;
4. \hspace{1em} if $i \geq j$ then
5. \hspace{2em} Forward $e$ to the coordinator;
6. \hspace{1em} else
7. \hspace{2em} Discard $e$;
8. \hspace{1em} end
9. end

Coordinator

Algorithm 2: SP at the coordinator in round $j$

1. foreach event $e$ do
2. \hspace{1em} $i \leftarrow \arg \min_{x \in \mathbb{N}} \{h(e) \geq 2^{-x-1}\}$;
3. \hspace{1em} $k \leftarrow e.\text{source}$;
4. \hspace{1em} if $i \geq j$ then
5. \hspace{2em} $Q^k_i.\text{add}(e)$;
6. \hspace{2em} $l'_i \leftarrow l'_i + 1$;
7. \hspace{2em} $g \leftarrow g + 1$;
8. \hspace{2em} while $g > B$ do
9. \hspace{3em} Discard queues $\forall k, Q^k_j$;
10. \hspace{3em} $g \leftarrow g - l'_j$;
11. \hspace{3em} $j \leftarrow j + 1$;
12. \hspace{3em} Broadcast $j$ to all sensing devices;
13. \hspace{2em} end
14. \hspace{1em} else
15. \hspace{2em} Discard $e$;
16. \hspace{1em} end
17. end
Evaluation

• Simulation and prototype emulation
  – Real dataset: 5 event-based sensing data

• Prototype emulation
  – DSA
  – MD5, SHA1, SHA256
Simulation

Uniformity

Uniformly drawn data reveals substantial information

2% data => 5% error

(a) Deviation from the ground truth

(b) Impact of budget limit
Prototype emulation

GSC is faster under all cases

Sign-each is more than 50X slower
Prototype emulation

![Graph showing events per second vs. space available at the device]
Prototype emulation
Prototype emulation

![Graphs showing performance metrics for different algorithms and budget sizes.](image)
Conclusion

• Requirement of IoT communication
  - Computation efficiency
  - Uniformity
  - Partial data retrieval

• GSC is able to satisfy all these three requirements simultaneously.
Thank you!