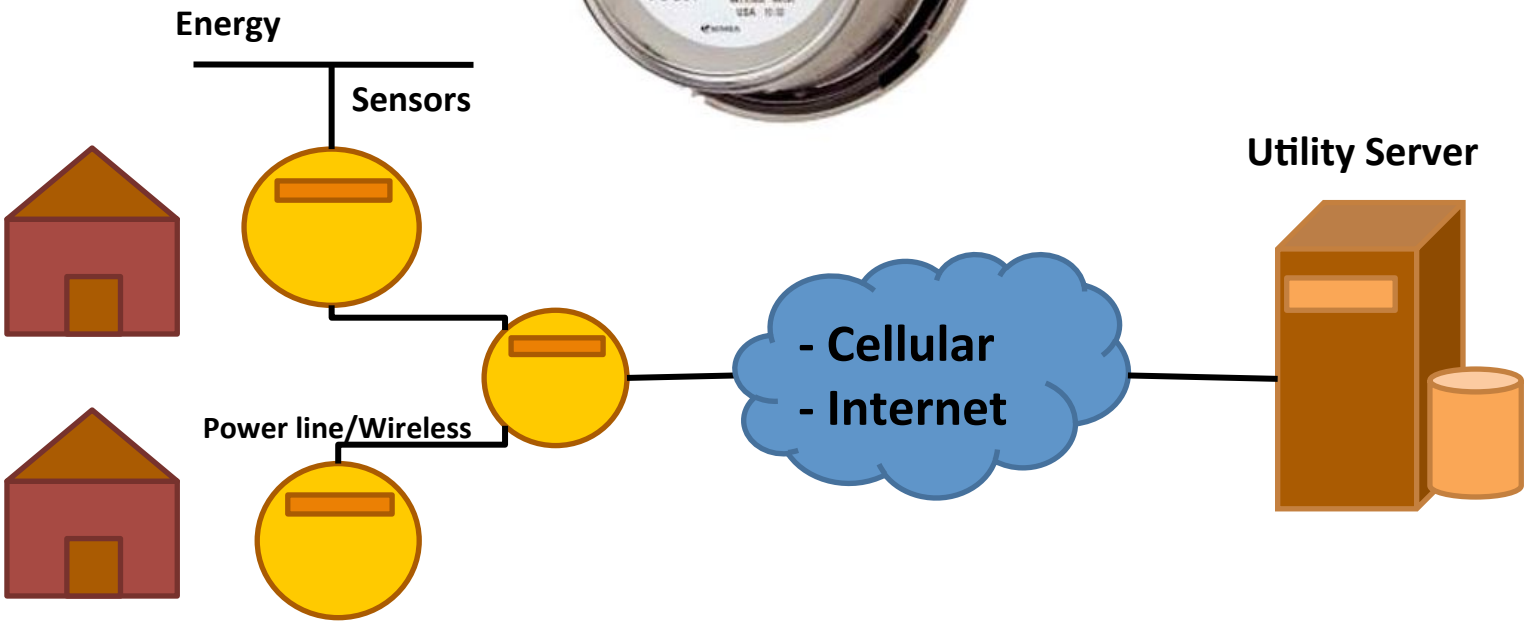
A Model-Based Intrusion Detection System for Smart Meters

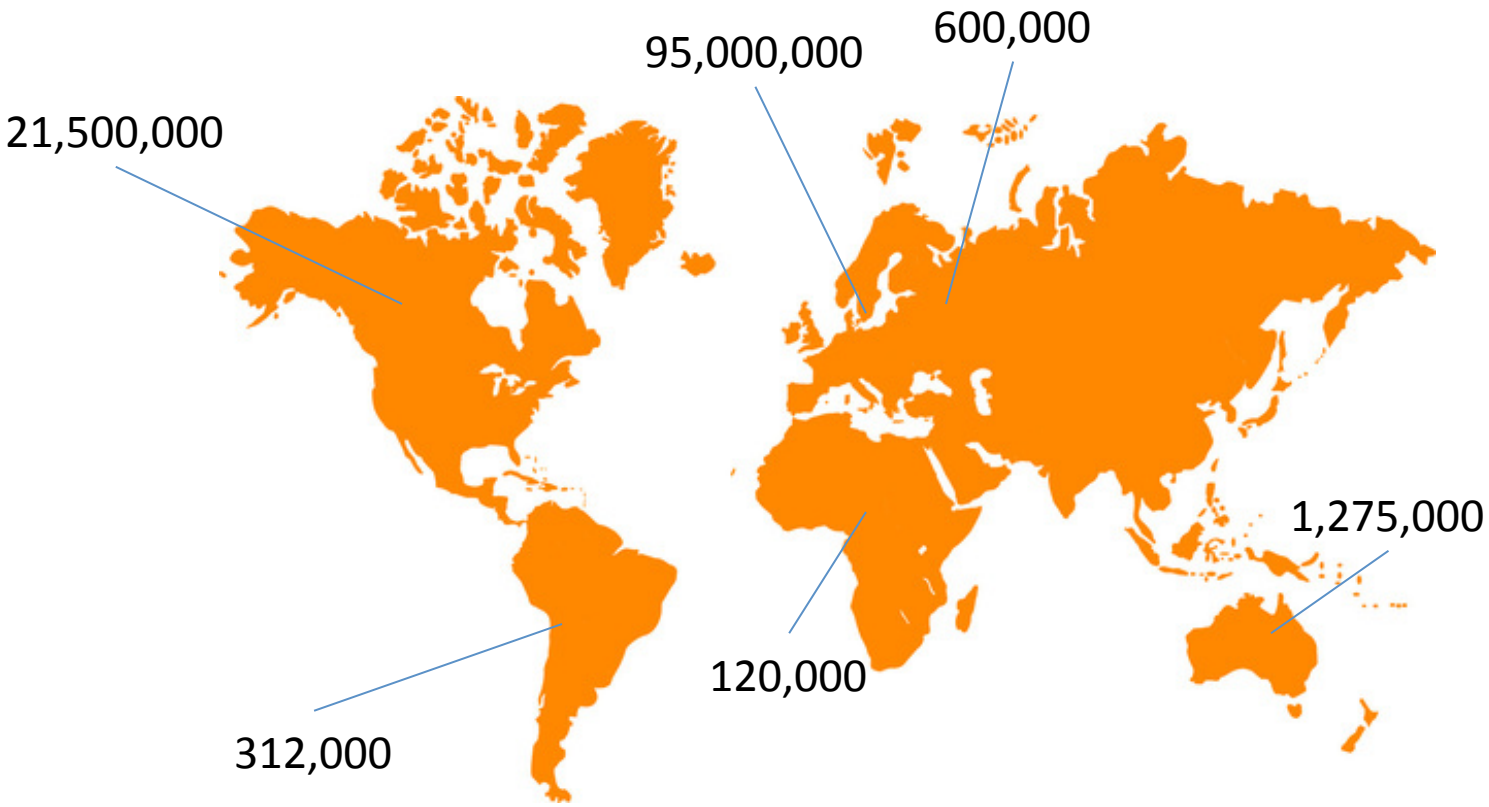
Farid Molazem Tabrizi
Karthik Pattabiraman

Dependable Systems Lab
University of British Columbia

Smart Meters

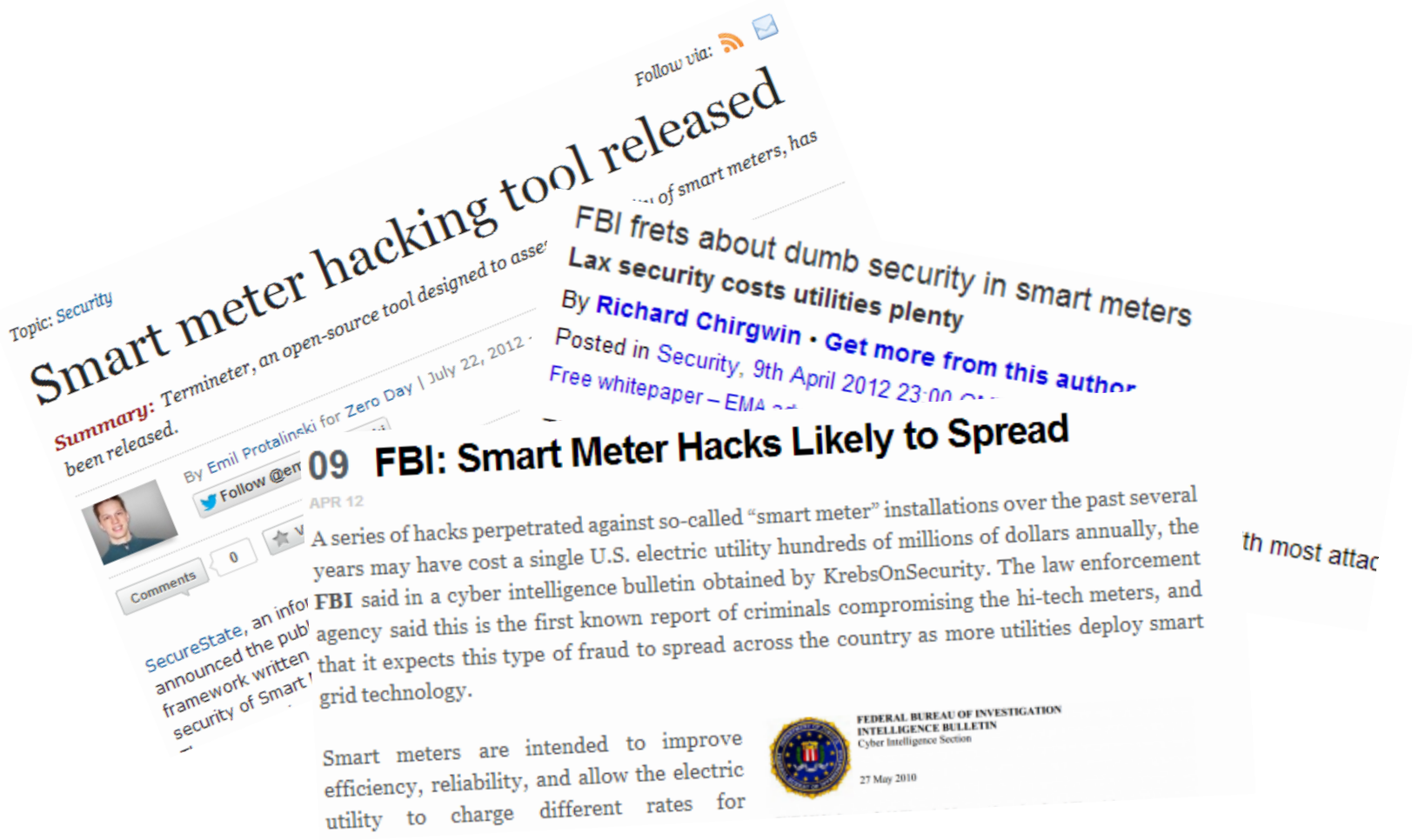


Global usage



By 2020: 1 billion

Security is a concern



Goal

- **Goal: Make smart meters secure**
  - Build a host-based intrusion detection system (IDS)

- **Why is it a new challenge?**
  - Smart meters have constraints that make them different from other computing devices
  - These constraints preclude existing IDS systems from running on them

Constraints of smart meters

- Memory & Performance constraints
- No false positives
- No software modification
- Low cost (no special hardware)
- Coverage of known attacks
- Coverage of unknown attacks

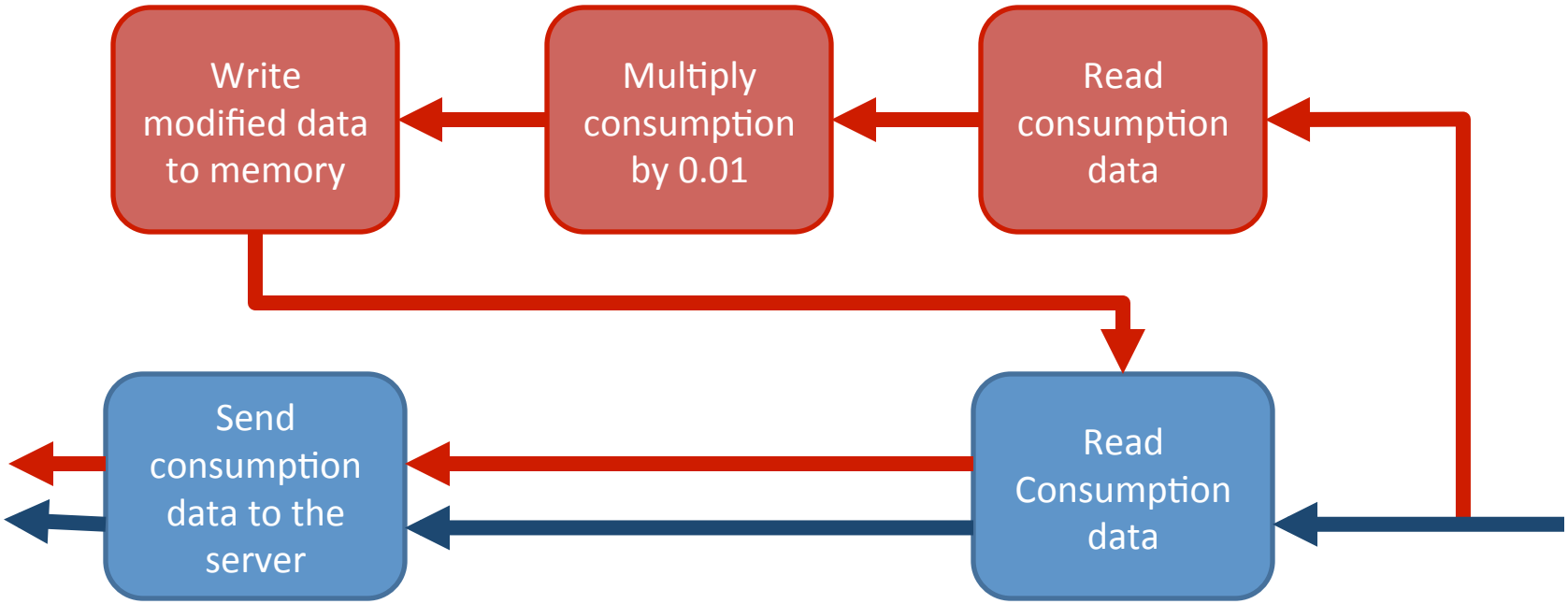
Prior Work on Intrusion Detection

System	Performance	False Positives	Software Modification	Low Cost	Known attacks	Unknown attacks
Dyck		X			X	X
NDPDA		X		X	X	X
HMM/NN/SVM	X		X	X	X	X
Statistical Techniques	X		X	X	X	X

No existing IDS can satisfy all six constraints: Need for new IDS

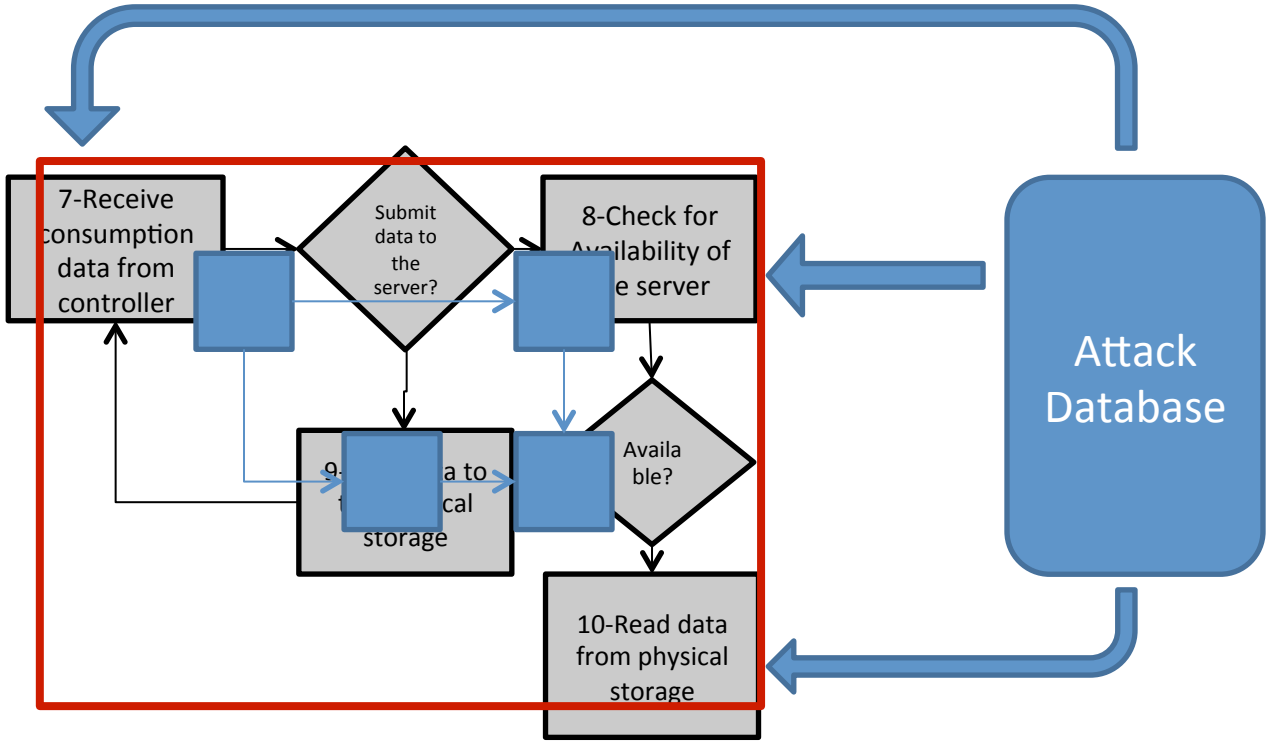
Threat model

- Adversary: wants to change the execution path of the software (maybe in a subtle way)

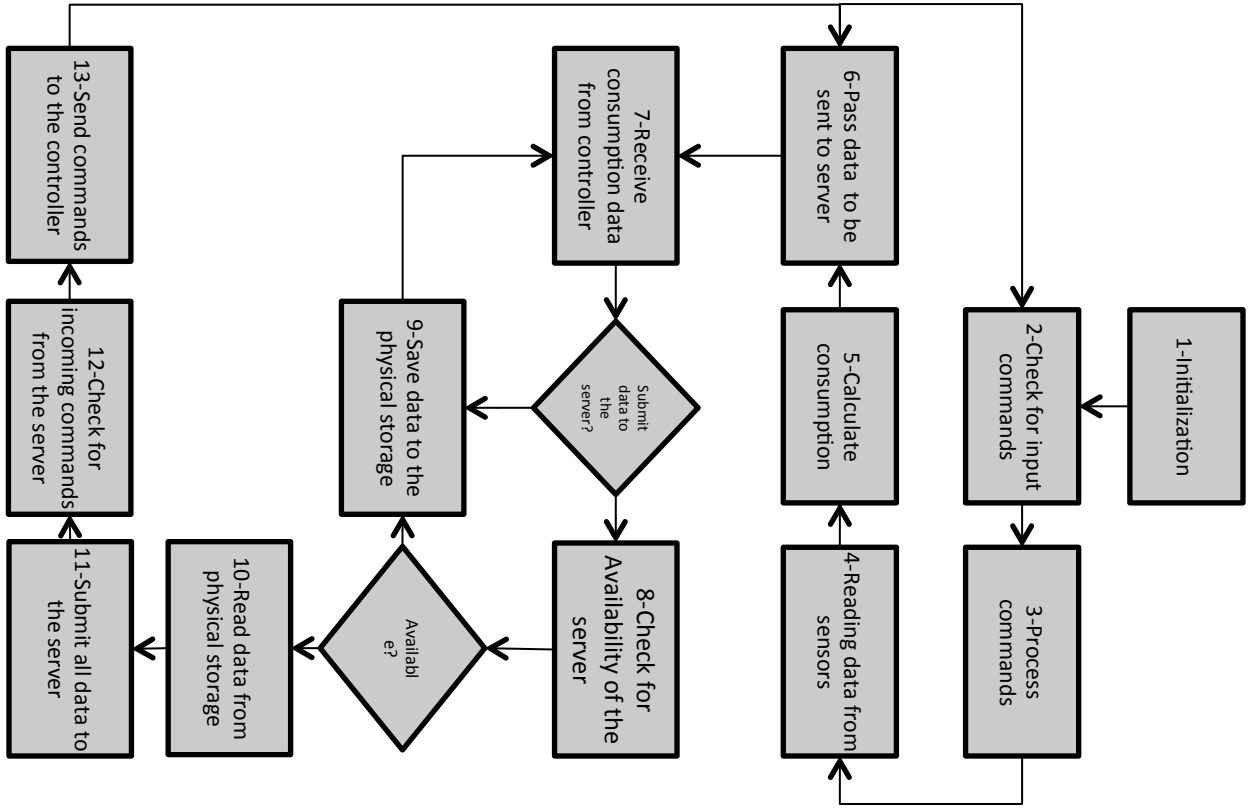


Our Approach

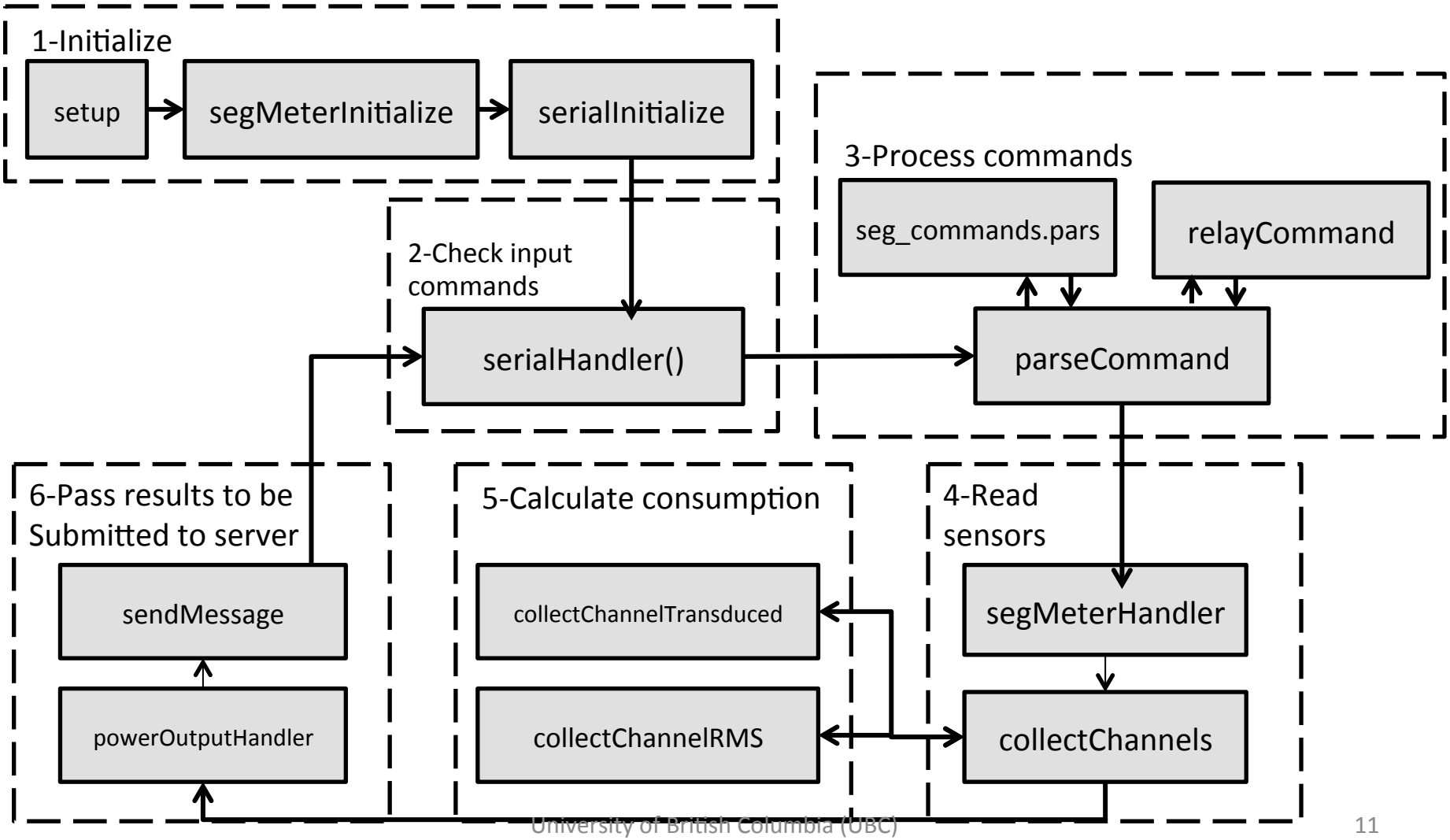
- Build the IDS based on model of smart meter



Abstract model: Based on Specification [WRAITS'12]



Concrete model: Based on Implementation



Building the concrete model

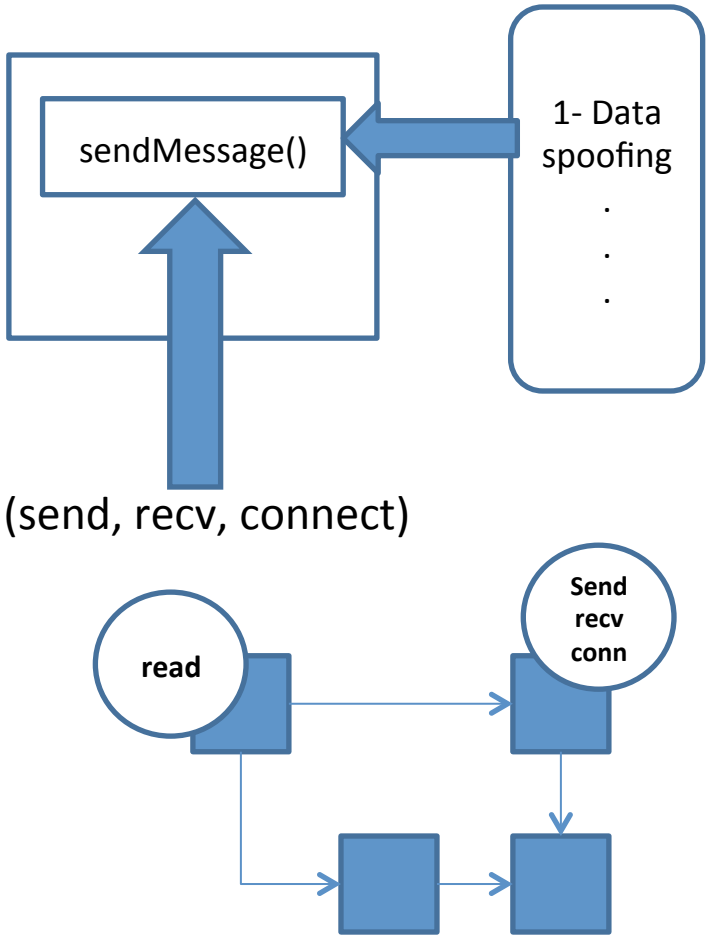
- Use a tagging system
- Tags defined based on abstract model

```
// <network, serial, b2>
SerialHandler()
{
…
}
```

- Features:
  - Ease of use
  - Flexibility

System call selection

- Generate the set of all system calls of the meter
- Traverse the attack database
- Map the attacks to specific blocks of the concrete model
- Pick system calls that cover the chosen blocks until all blocks are covered
- Generate the state machine of the system calls based on the resulting graph



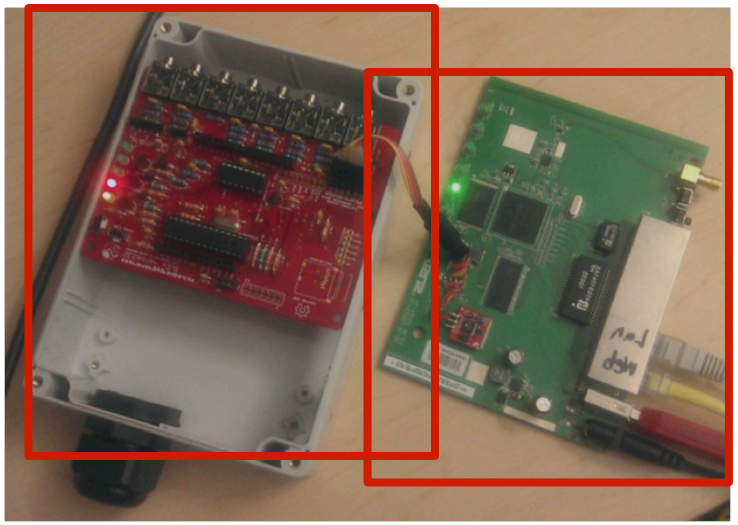
IDS Implementation

- **Offiline: Generate state machine of system calls**
  - Input: system call patterns
  - Output: state machine

- **Online: Two components**
  - **Logger**: Attaches *strace* to the process being monitored and logs system call traces
  - **Checker**: Runs every '*T*' second, parses the generated system calls, checks the logged trace with the model

Evaluation

- **SEGMeter**
  - Arduino board
    - ATMEGA 32x series
    - Sensors
  - **Gateway board**
    - Broadcom BCM 3302 240MHz CPU
    - 16 MB RAM
    - OpenWRT Linux
  - IDS runs on Gateway board



Results: Performance

- Performance
  - The ratio of the time taken to check the syscall trace, to the time taken to produce the trace

Memory available	12 MB	9 MB	6 MB
Full-trace IDS	165.2%	214.6%	315.1%
Our Model-based IDS	4.0%	4.0%	4.0%

Full-trace IDS cannot keep up with the software, while our model-based IDS incurs low overheads

Results: Coverage (Unknown Attacks)

- **Detection (Unknown attacks)**
  - Code injection
    - Select a procedure to inject in the smart meter
    - Mutate the procedure by copying and pasting 1-8 lines of code from some other part of it (makes it harder to detect)

Our Model-Based IDS achieves nearly the same coverage as the full-trace IDS for a fraction of the cost, and significantly higher coverage than the random and "most popular calls" IDSes.

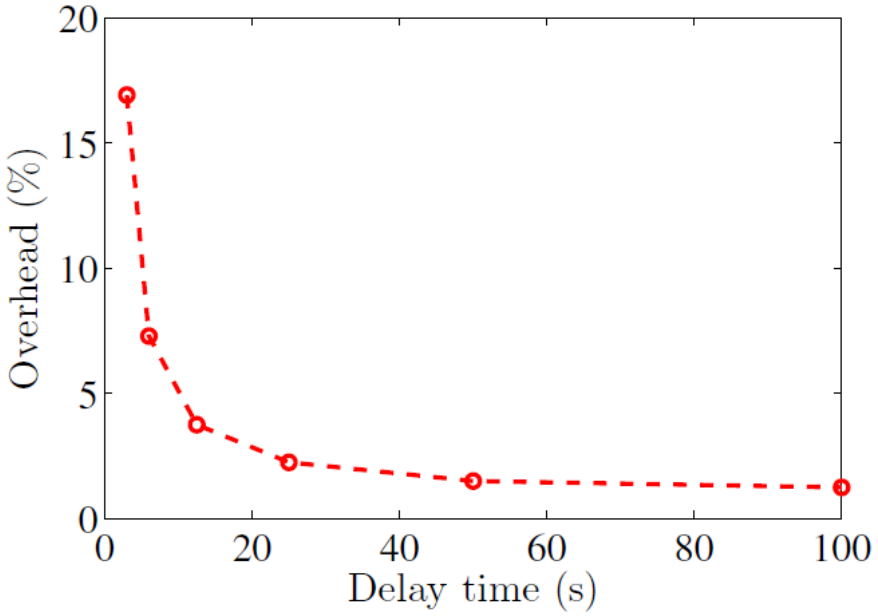
Component	Selection (%)	Parameter-value calls (%)	Sub-trace (%)	Methods Chosen		
				Minimum	Average	Maximum
Server communication	32	36	92	59	62	63
Storage and retrieval	14	44	84	73	74	78
Serial communication	42	28	88	67	72	74
Total	29.3	36.0	88.0	67.4	69.6	71.7

Results: Coverage (Known Attacks)

- **Detection (Known attacks)**
  - Implemented four different attacks [WRAITS'12]
    - Communication interface attack
    - Physical memory attack
    - Buffer filling attack
    - Data omission attack
  - **Our Model-Based IDS detects all four attacks**
    - If undetected, the attacks lead to severe consequences

Results: Monitoring Latency

- Monitoring latency
  - Smaller *T*: Faster detection, higher performance overhead
  - We pick *T*= 10s
    - Low performance overhead: 4%
    - Fast detection



Conclusion

- **Smart meters have special constraints that *cannot* be met with existing IDSes**
- **Our model-based IDS: Based on smart meter's requirements and high-level model of operation**
  - Low performance overhead under memory constraints
  - Good detection coverage of known and unknown attacks
- **Future work:**
  - Generalize to other smart meters
  - Formalize model and IDS specifications