Leveraging Blockchain-based protocols in IoT systems

Angelos Stavrou

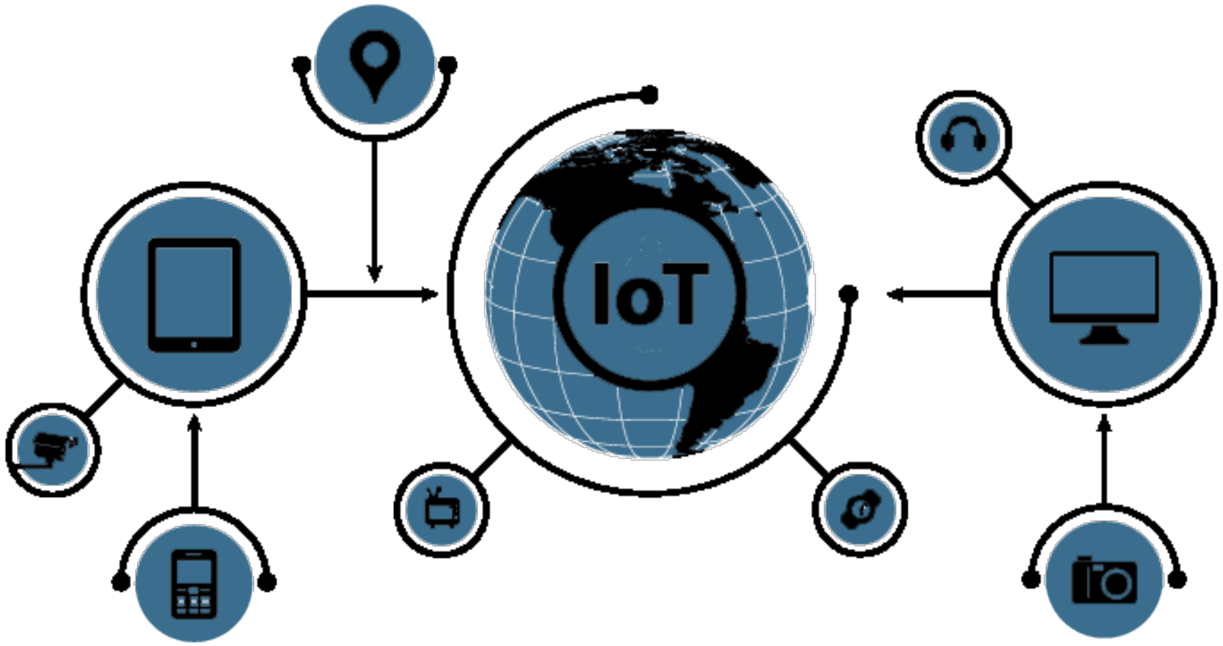
George Mason University | Volgenau School of Engineering

Talk Outline

- Overview of IoT
- Security Failures in IoT: Motivating Use Cases
- **Why direct use of Blockchain is not practical for IoT**
- **Challenge**: Design practical Blockchain-based protocols for IoT
- Conclusions, Discussion & Challenges

Internet of Things Defined

- Kevin Ashton introduced the term Internet of Things (IoT) in 1999
- Network of devices able to configure themselves automatically
- Human is not the center of the system
- **Motivation**: Better understanding of the environment and response to certain events. Machines are doing better in sensing & reporting on conditions
- **Fact**: Applications of traditional Internet are different than the applications of IoT



Cyber Security is not a Design Tenet

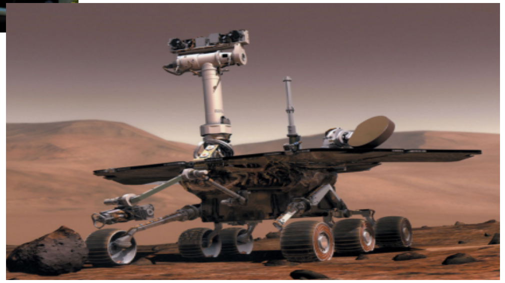
What is the Fundamental Problem?

- Devices operate using **non-verified or tested software**
  - outdated software
  - custom-made software
  - software from many vendors
  - modular software from **many different vendors**
  - poorly tested software
  - software that was designed for a different set of requirements
  - unpredictable & chaotic software

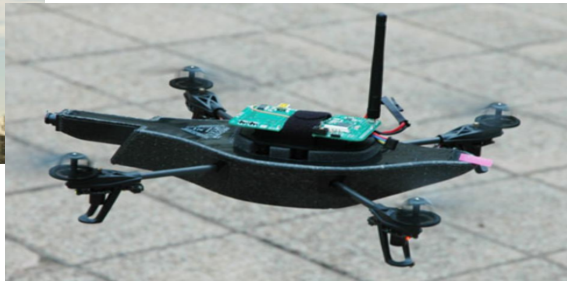
There is NO Industry incentive to build Secure Systems (Software or Hardware)

What the Future Holds

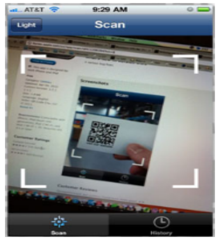
Drivables



Flyables

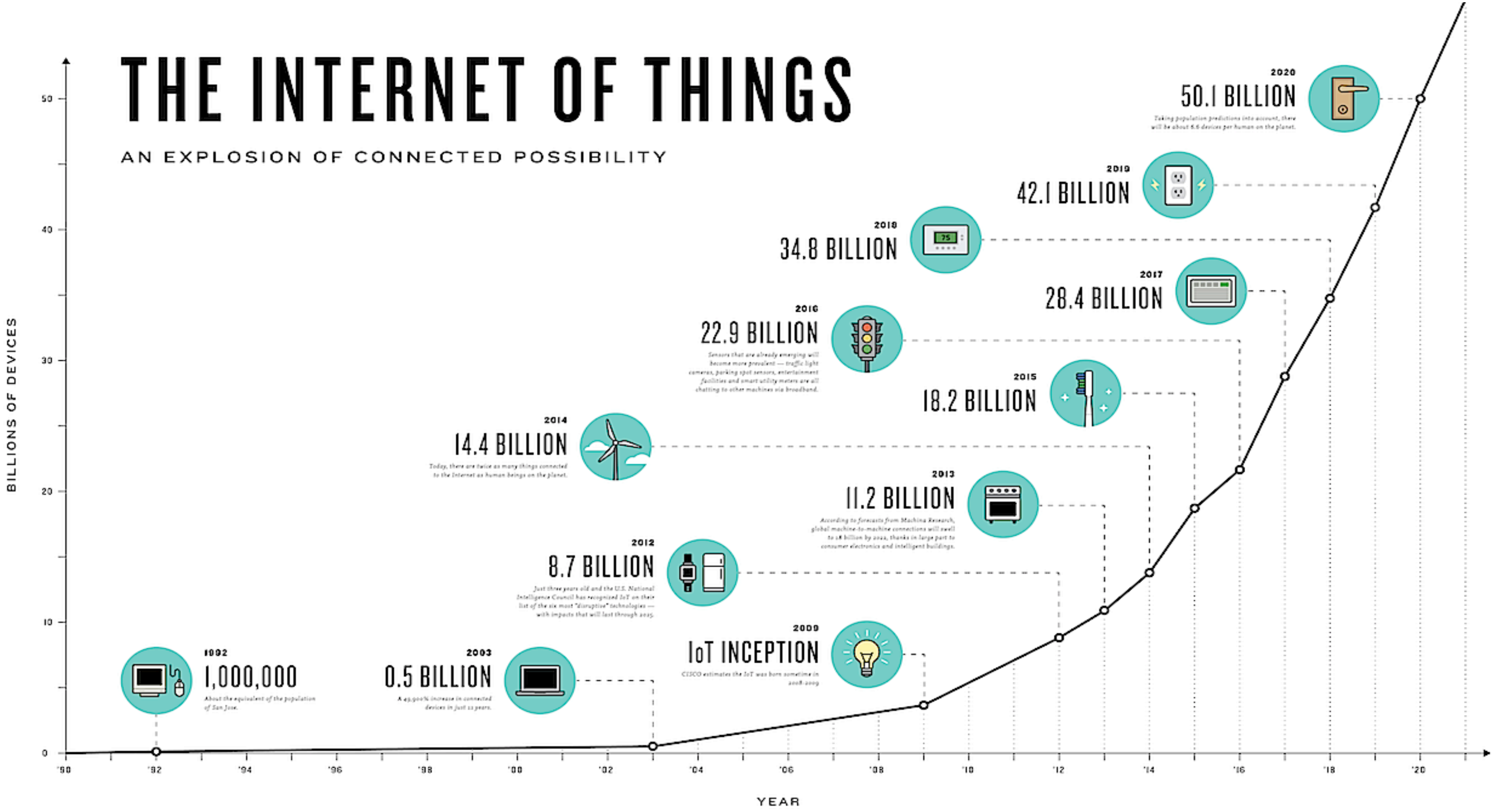


Scannables



Wearables

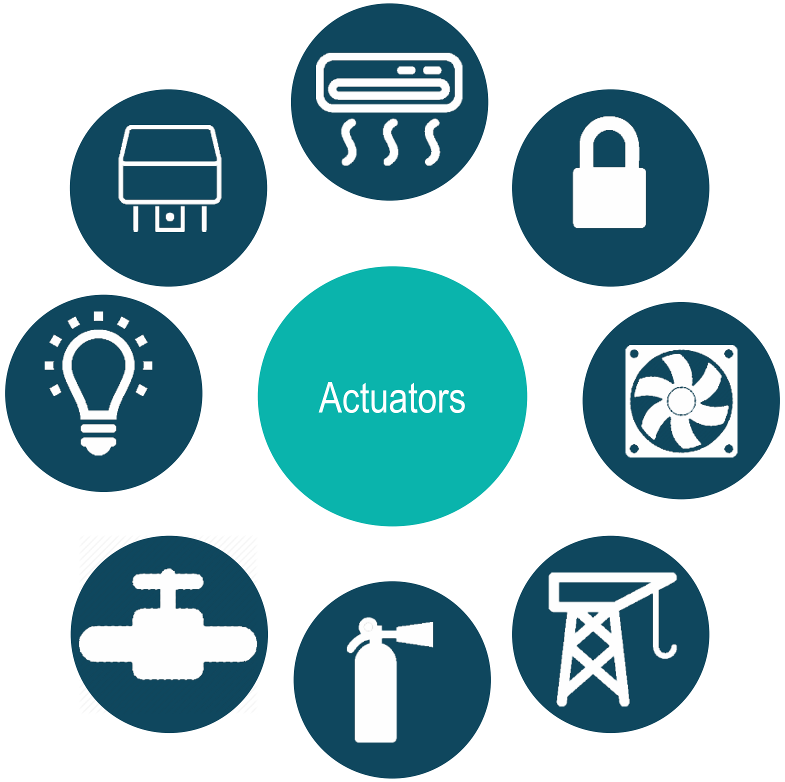
The Growth of IoT



Sectors of IoT Applications

Smart Home	Transportation	Retail	Industry	Healthcare
Home automation	Road safety	Automatic payments	Quality assurance	Condition monitoring
Energy efficiency	Traffic regulation	Efficient cataloguing	Failure prediction	Remote treatment
Home security	Law enforcement	Shipment tracking	Productivity improvement	Personalized advices

Sensors & Actuators



Connectivity

Talk Outline

- Overview of IoT
- Security Failures in IoT: Motivating Use Cases
- **Why direct use of Blockchain is not practical for IoT**
- **Challenge**: Design practical Blockchain-based protocols for IoT
- Conclusions, Discussion & Challenges

Common Security Incidents

Private Data Collection | **Insecure Interfaces** | **Unencrypted Communications** | **Weak Requirements**

Top 10 Vulnerabilities (OWASP)

- Insecure Web Interfaces
  Default accounts, XSS, SQL injection
- **Inefficient Authentication/Authorization**
  Weak passwords, no two-factor authentication
- Insecure Network Services
  Ports open, use of UPnP, DoS attacks
- **Lack of Transport Encryption**
  No use of TLS, misconfigured TLS, custom encryption
- Private Data
  Unnecessary private information collected
- Insecure Cloud Interfaces
  Default accounts, no lockout
- **Inefficient Mobile Interfaces**
  Weak passwords, no two-factor authentication
- Insufficient Security Configurability
  Ports open, use of UPnP, DoS attacks
- Insecure Software/Firmware
  Old device firmware, unprotected device updates
- **Poor Physical Security**
  Exposed USB ports, administrative accounts

Use Case: Bluetooth Low Energy Beacons

- Beacons Purpose:
  - Provide inexpensive remote identification
  - Proximity estimation
  - Low power consumption

- BLE modules are integrated with smartphone devices

- Hardware requires very little energy
  - Easy to maintain and have a small footprint

- Achieve accurate proximity estimation even in indoor scenarios
  - Better than GPS

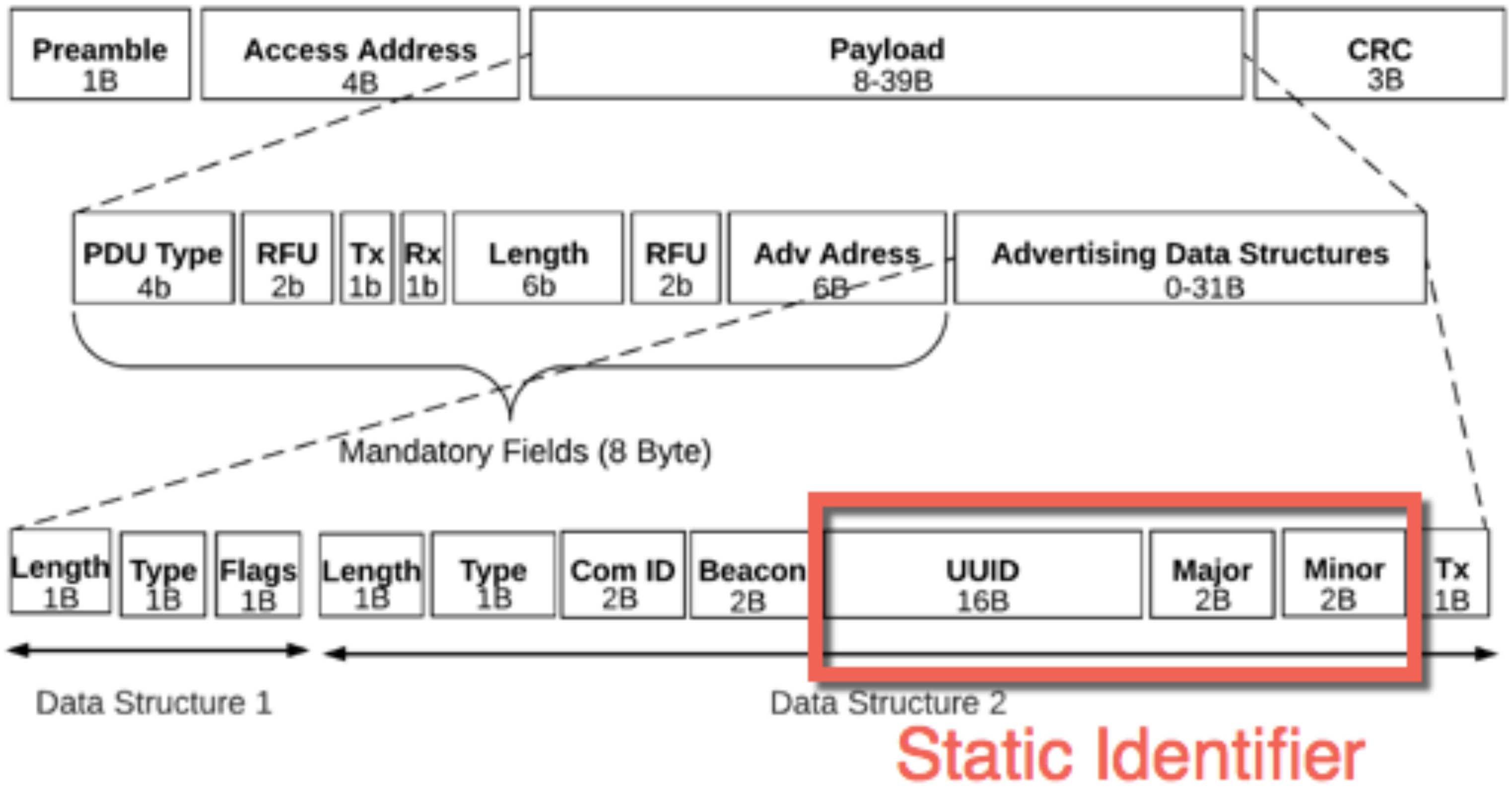
- Identification can be achieved across considerable distances
  - Better than RFID

What Can Go Wrong?

- Existing BLE Beacon specifications naively omit protection in message structure
  - Apple's iBeacon, Google's Eddystone, Altbeacon
- Vendors claim that BLE Beacon applications are not security & privacy sensitive
- Current Applications can be abused
  - Denial of service or loss of revenue
- What about future applications?
  - Automatic payments
  - Automatic Check-In
  - Authorization to Restricted Areas
  - Access control to devices (e.g. workstation)

Underlying Design Problem

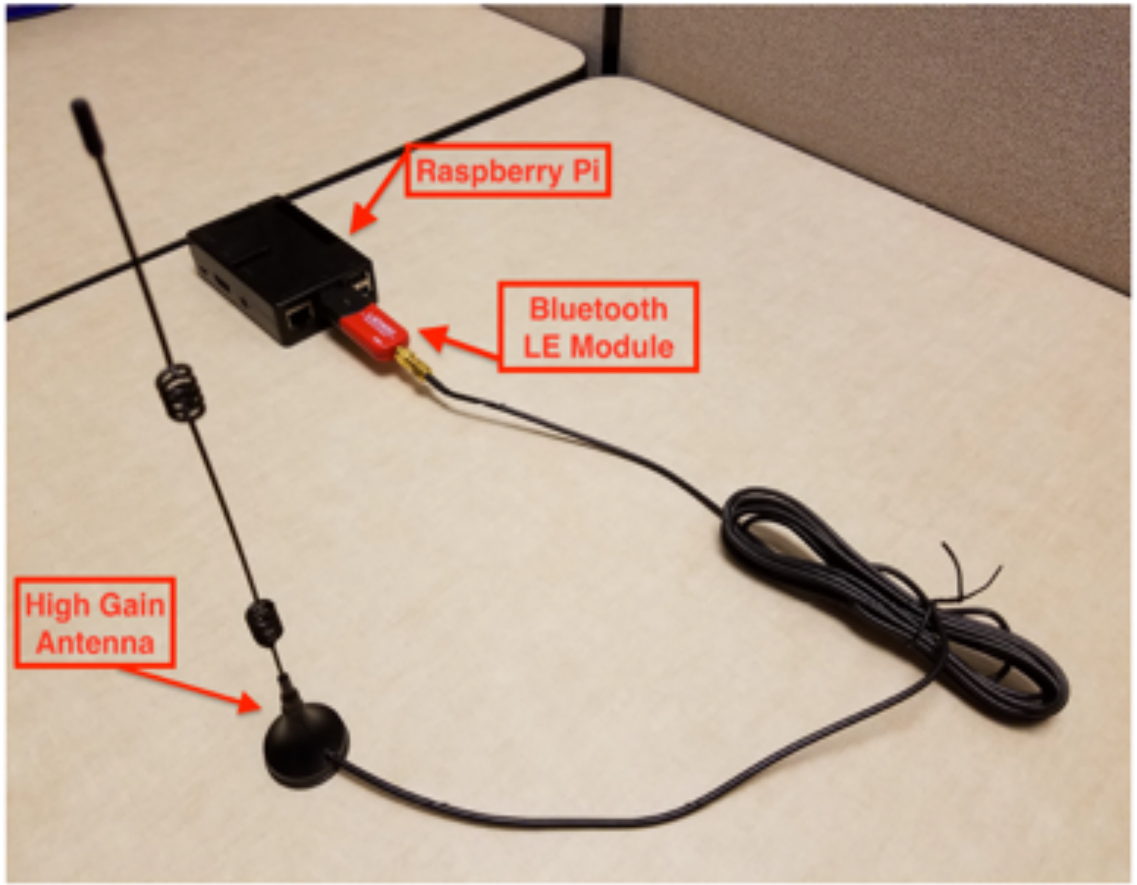
- Transmission of a static identifier
- Constant broadcasting of that identifier
- Long range transmissions (75 meters)



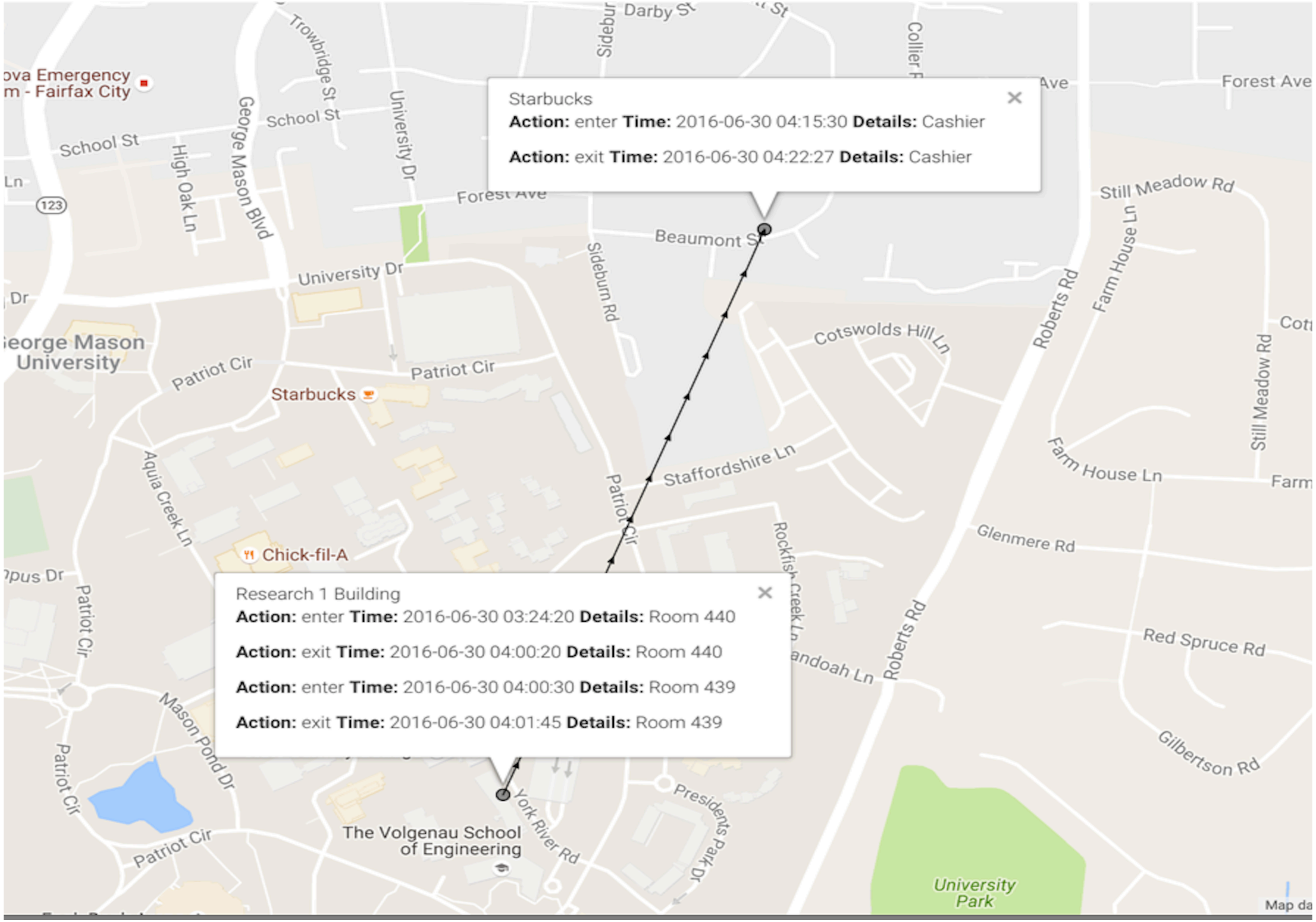
Attacker Capabilities

- Open source software for monitoring
  - Bluez, Ubertooth, others

- Inexpensive hardware
  - USB adapter (Sena UD100 Long Range Bluetooth 4.0 Class1 USB adapter)
  - High gain antennas (RP-SMA 2.4GHz 7 DBI)
  - Discrete portable devices (e.g. Raspberry Pi)

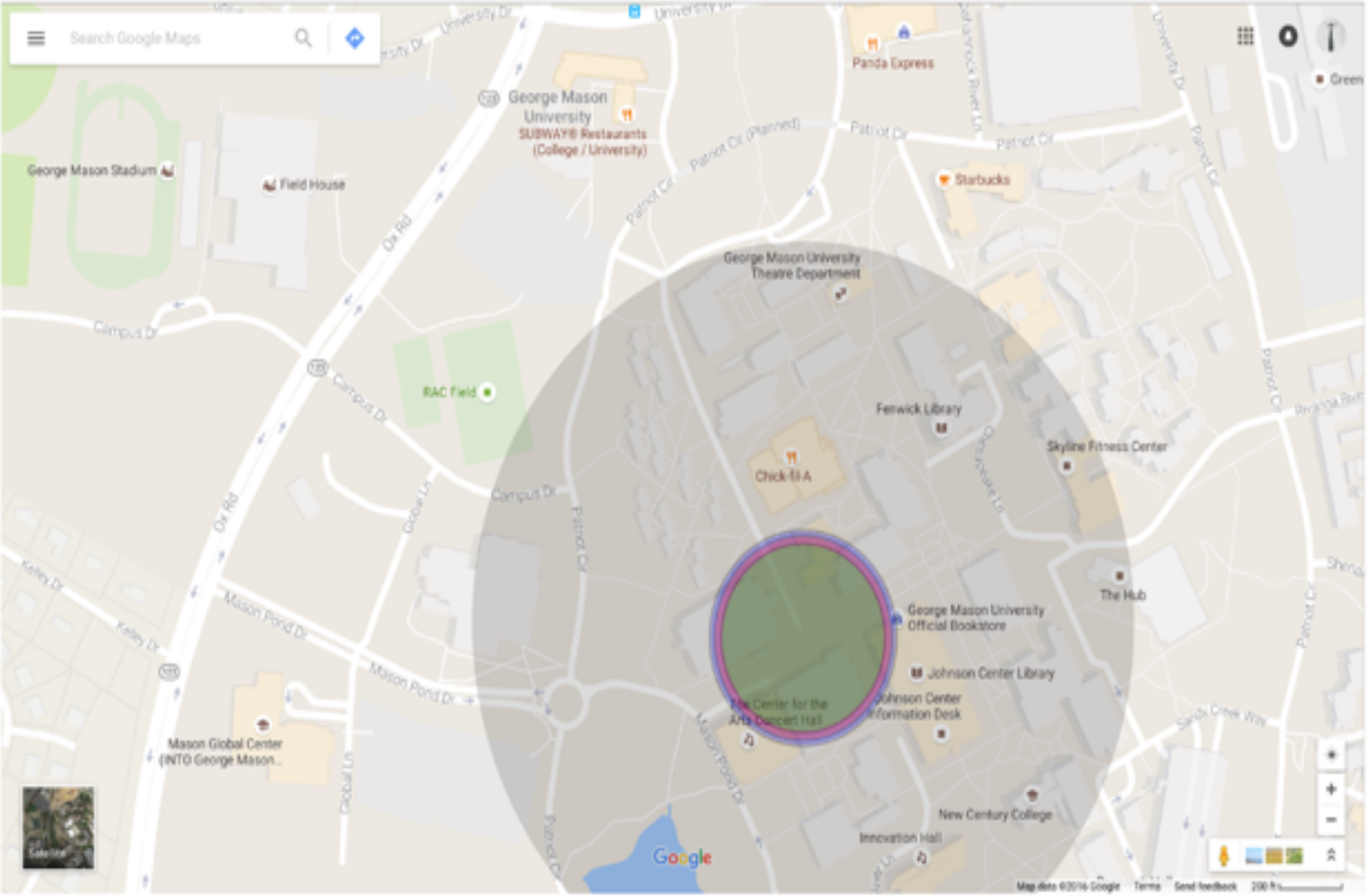


Attack: User Profiling



Attack: Presence Inference

- Tracking & Reporting the presence of a target within an area
- Target must carry a portable, beacon-emitting object
- Inexpensive equipment can boost the range to more than 300 meters radius
  - Typical range is 75 meters



Why not Use Cryptography?

GEORGE MASON UNIVERSITY

RSA 1024 Runtime Overhead:

Arduino UNO	16Mhz AVR	==>	12596 ms*	8504 ms#
Arduino Leonardo	16Mhz AVR	==>	12682 ms*	8563 ms#
Arduino Mega	16Mhz AVR	==>	12596 ms*	8504 ms#
Arduino Due	84Mhz ARM	==>	1032 ms*	
Arduino Yún	16Mhz AVR + 400Mhz MIPS	==>	707 ms*	
Intel Galileo	400Mhz x86	==>	192 ms*	

* these numbers are based on a 100% C implementation

these numbers are based on mixed C/AVR assembly implementation

Some of the traditional Crypto is too “expensive” for embedded devices

Survey of Crypto Support in IoT

Brand	Name	CPU	Freq.	Sram	Flash	Crypto Acc.	Energy Source	Public Key Crypto
Belkin	WeMo Switch	Ralink RT5350F (MIPS)	360 Hz	32MB	16MB	No	Wall socket	Yes
Samsung	Smarthings Hub	PIC32MX695F-512H	80MHz	128KB	512K	No	Wall socket/Battery	Yes
Nest	Thermostat	TI AM3703CUS Sitara (ARM Cortex A8)	1GHz	512Mb	2Gb	Yes	Wall socket	Yes
LIFX	Color 1000	Kinetis K22 (ARM Cortex-M4)	120MHz	128KB	512K	No	Wall socket	No
Amazon	Echo	TI DM3725CUS100 (ARM Cortex A8)	1GHz	256MB	4GB	Yes	Wall socket	Yes
Philips	Hue Lights	ST Mic. STM32F217VE (ARM Cortex-M3)	120MHz	128KB	1MB	Yes	Wall socket	Yes
Philips	Hue Lights (Bulb)	STM32F100RBT6B (ARM Cortex-M3)	24MHz	8KB	128KB	No	Wall socket	No
Nest	Smoke/Carbon Alarm	Freescale SCK60DN512VLL10 custom Kinetis K60	100MHz & 48MHz	128KB	512K	Yes	Wall socket/Battery	Yes
Pebble	Time	ST Micro STM32F439ZG (ARM Cortex M4)	180MHz	256KB	2MB	Yes	Battery	No
Adafruit	Feather MO Bluefruit LE	TSAMD21G18 ARM Cortex M0	48MHz	32KB	256KB	No	Battery	No
BeagleBone	Green Wireless (other models)	AM335x 1GHz ARM Cortex-A8	1GHz	512MB	4GB eMMC	Yes	External/Battery	Yes
Raspberry Pi	Zero	ARM1176JZFS Armv6 core	1GHz	512MB	MicroSD	Yes	External/Battery	Yes
Raspberry Pi	Two (2)	ARM Cortex-A7	900MHz	1 GB	MicroSD	Yes	External/Battery	Yes
Raspberry Pi	Three (3)	ARM Cortex-A53	1.2GHz	512MB	MicroSD	Yes	External/Battery	Yes
Arduino	MKR1000 (other models)	Atmel | SMART SAMD21 Cortex-M0+	32KHz & 48MHz	32KB	256KB	No	Battery	No
Fitbit	One	ST Mic. 32L151C6 Ultra Low P. ARM Cortex M3	32 MHz	16KB	128KB	No	Battery	No
Fitbit	Surge	Silicon Labs EFM32 (ARM Cortex-M3)	48 MHz	128KB	1MB	Yes	Battery	No

Talk Outline

- Overview of IoT
- Security Failures in IoT: Motivating Use Cases
- **Why direct use of Blockchain is not practical for IoT**
- **Challenge**: Design practical Blockchain-based protocols for IoT
- Conclusions, Discussion & Challenges

Can we use Blockchain-inspired protocols?

Strengths

- Trust among untrusted Parties
- Distributed resilience and control
- Fully Decentralized network
- Primarily Open source
- Security and modern cryptography
- Controlled & Open Participation
- Smart Contracts
- Dynamic and Fluid Operation

What do we **really** need?

IoT System Operational Requirements (Empirical)

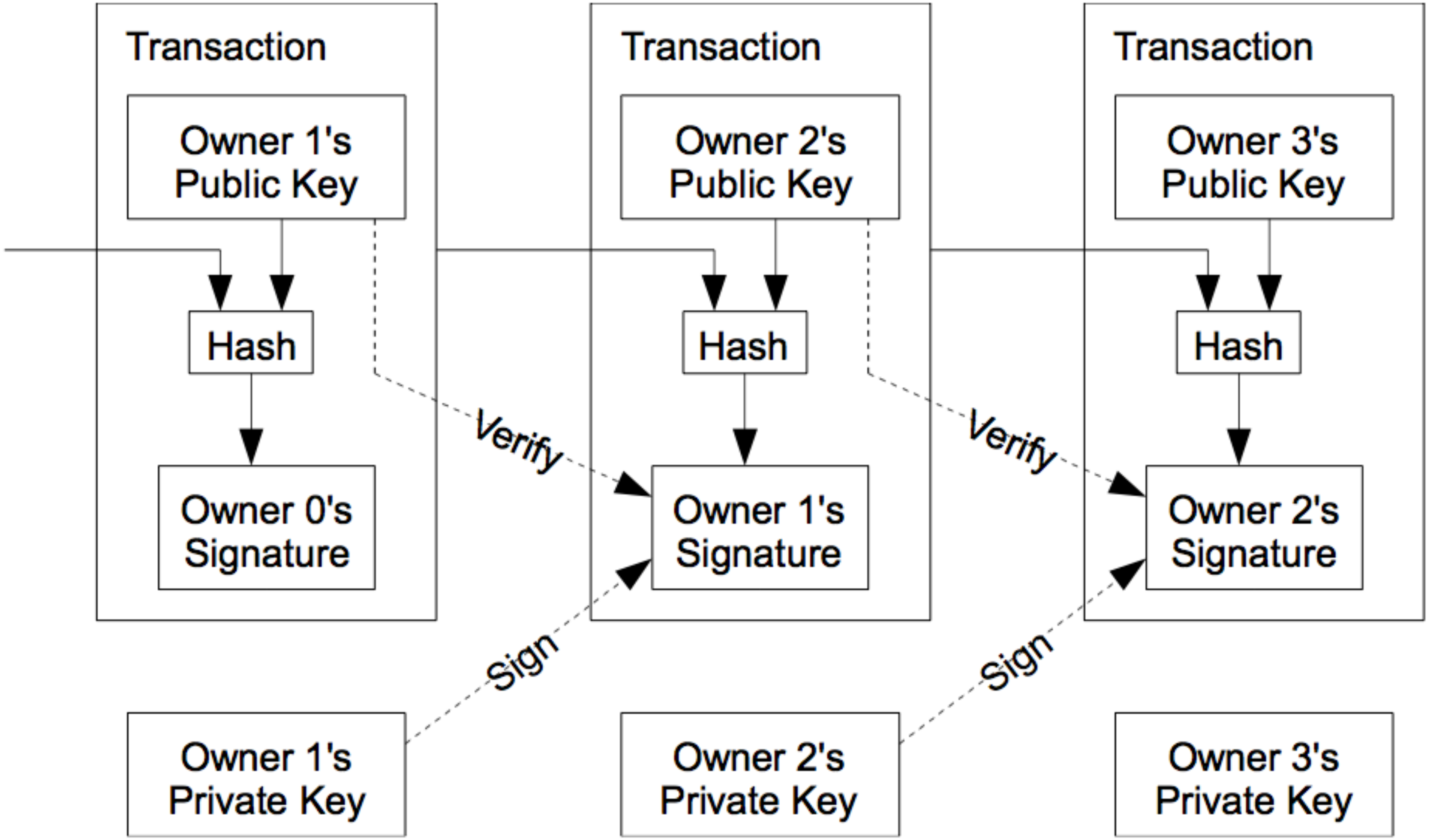
- Dynamic but verifiable group membership
- Authentication & Data integrity
- Secure against single-node (or small sub-set of nodes) key leakage
- Lightweight operations in terms of resources
- Encryption is a plus but not firm requirement
- Capable of handling sensor "sleep/power-off" periods
- Handle resource diversity and data of sensors and aggregators

Blockchain Primer

Public Distributed Verifiable Cryptographic Leger

- Public
  - All participants gain access to “read”
- Distributed
  - Peer-to-Peer Data Communication, Fully Decentralized
- Cryptographic
  - Digitally signed transactions, proof-of-work limits rate of input
- Ledger
  - Verifiable Transactional Database

Blockchain Primer

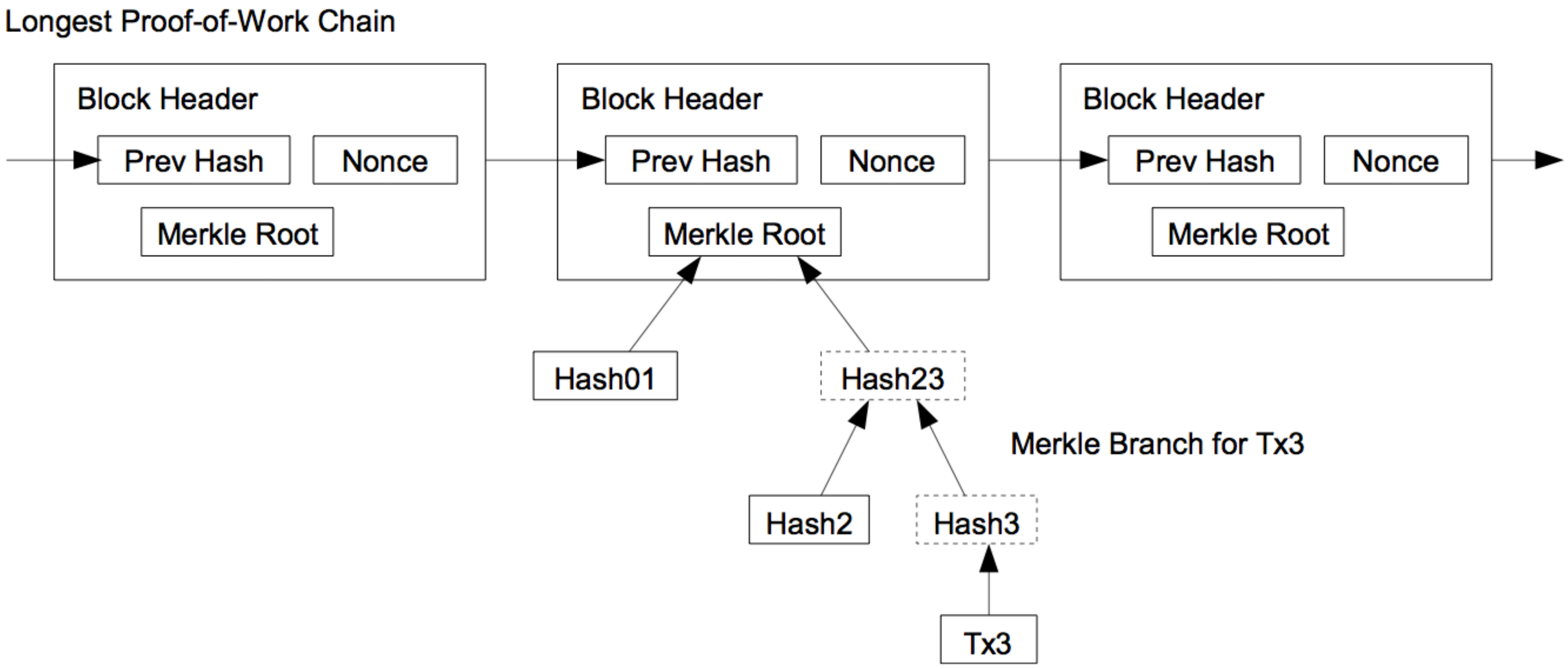


Blockchain Primer

Blockchain Blocks

- Sequences of signed and verified transactions
- Published and distributed globally
- Magic number, Size
- Header
  - Hash of previous block (chain)
  - Merkle root hash of block
  - Timestamp
  - Target, nonce (mining)
- Number and list of transactions

Blockchain Primer



Talk Outline

- Overview of IoT
- Security Failures in IoT: Motivating Use Cases
- **Why direct use of Blockchain is not practical for IoT**
- **Challenge**: Design practical Blockchain-based protocols for IoT
- Conclusions, Discussion & Challenges

Is Blockchain Directly Applicable in IoT?

Desirable Properties

- Distributed protocol with verifiable transaction history
- Dynamic membership multi-party signatures

Undesirable Properties

- Requires proof of "work"
- Requires PKI
- Size of the Ledger an issue for "small" devices
- Anonymous (unverifiable) Join/Leave operations

What can we do?

Eliminate undesirable properties

- ~~Requires proof of "work"~~

  Requires proof of earlier participation using history

- ~~Requires PKI~~

  Hash-based signatures (or other Merkle-tree schemes)

- ~~Size of the Ledger an issue for "small" devices~~

  Prune and Compress Ledger. Maintain only device-relevant transaction ledger when device is too resource constrained

- ~~Anonymous (unverifiable) Join/Leave operations~~

  Group signatures using pre-shared group Key(s)

Hash-Chains

One-time hash passwords (Lamport 1981):

- Client generates iteratively a list of hash values (in reverse order of index).

$$
\begin{aligned}
z_\ell &\leftarrow \{0,1\}^n \\
z_i &\leftarrow h(z_{i+1}) \quad \text{for } i \in \{\ell-1, \ell-2, \ldots, 0\}
\end{aligned}
$$

- $z_0 = h(z_1) = h(h(z_2)) = \ldots$ is the “public key”
- Keys are revealed in opposite order, starting from z_1
- Verification of z_i: starting from z_i verify, if z_0 is indeed i-th hash
- Keys can be used only once!

Hash-Chain: PreImage Path

Lamport's one-time-password scheme has either

- $O(\ell)$ storage (whole chain retained) or
- $O(\ell)$ preimage generation time (only z_ℓ retained).

Both extremes are not exactly efficient.

Naive optimization: mark few elements with "pebbles", retain values and use as starting points. If N pebbles are evenly distributed then the worst case is $O(\ell/N)$ hash calculations per key.

Jakobsson (2002): traversal algorithm which amortizes $h()$ calculations. $O(\log \ell)$ memory and $O(\log \ell)$ hashing steps to output a key (preimage).

Pebbles are placed at positions $2^j, j = 1..\lfloor \log \ell \rfloor$; preimages are extracted from left. If a pebble is reached it jumps next to another, and leftover calculations at each step are used to move it gradually into position between neighbors.

Hash-Chain: PreImage Cost

But what about in practice?

For sensor nodes and aggregators:

Using Hash chain of size: 2^{32} = 4,294,967,296 passwords

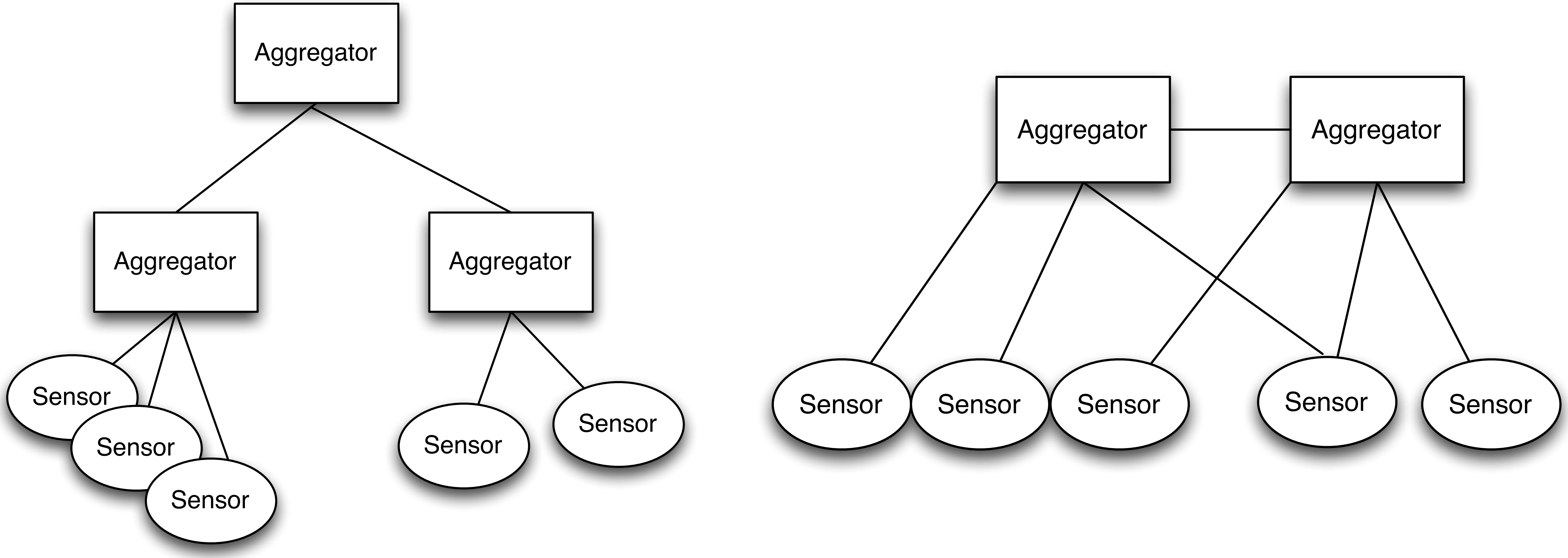
- More than **68 years** to run out for one (1) transaction per second
- Each transaction having a distinct key

If we select SHA256 as the hash function of choice:

Memory Requirements: $2 \times \log_2(n) + 256 = 320$ bits

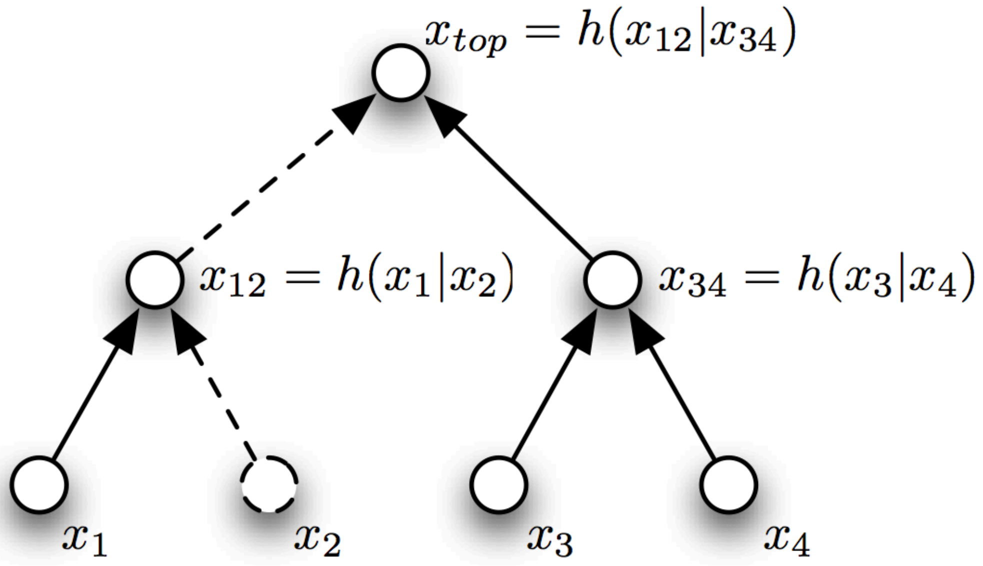
For 32 locations + seed totaling **1,320 bytes** of storage or **1.3KB**

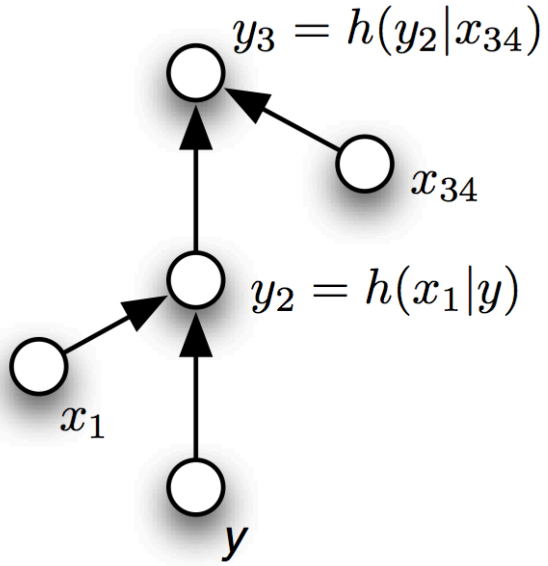
Typical Sensor Networks



Blockchain-based Protocol for IoT?

We suggest a Blockchain-based protocol that uses the following blocks:



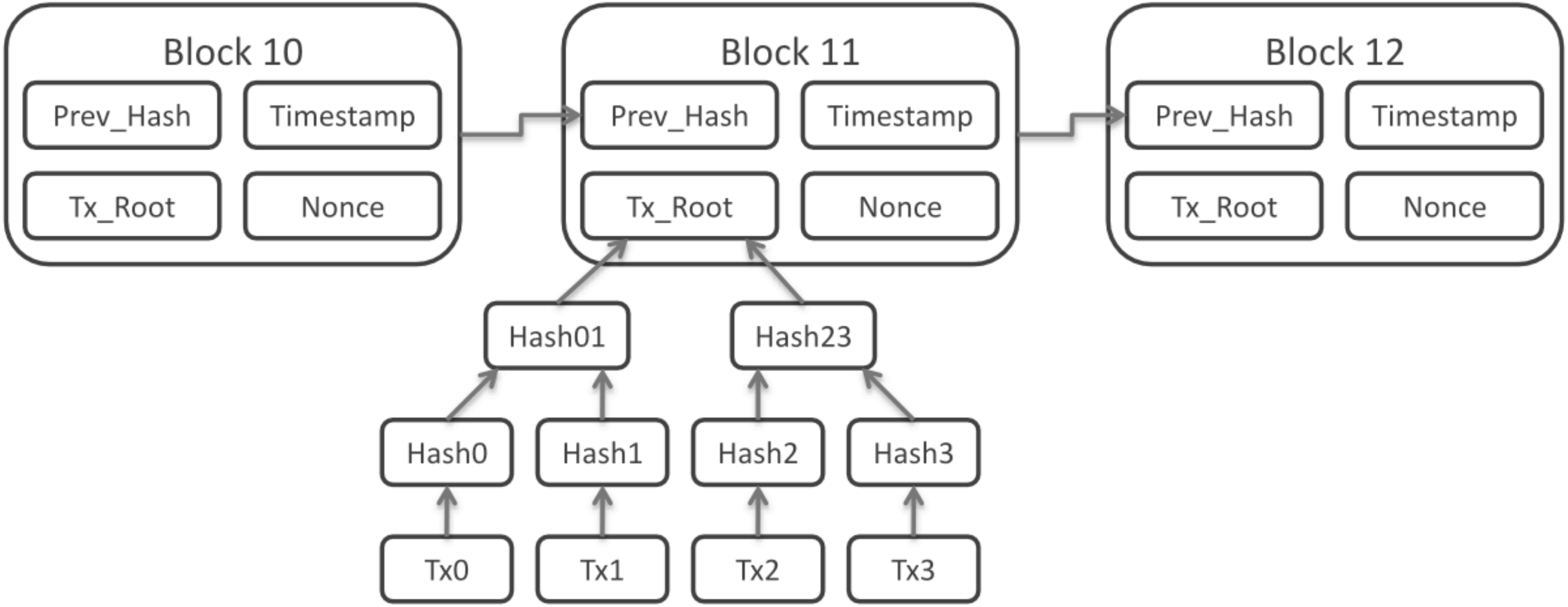


$$x_i = H(Data \,||\, K_G \,||\, H(z_i)^n), H(z_i)^{n-1}$$

$$H = Hash, K_G = group\ Key, z_i = sensor\ i\ "public\ \ key"$$

Blockchain-based Protocol for IoT?

We suggest a Blockchain-based protocol that uses the following blocks:



Does the Scheme Meet the Requirements?

- IoT System Operational Requirements (Empirical)
  - Dynamic but verifiable group membership
  - Secure against single-node (or small sub-set of nodes) key leakage
    - **Only Aggregators** can add nodes by issuing a group Key
    - Can be done using Symmetric Encryption or a Hash Chain
    - Node is verified both by **group key AND** by **participation history**
    - To add a node, an adversary will have to:

      a) Compromise the group key

      b) Issue an “add node” transaction

      c) Add a sensor node

    - Shape of the tree shows “additions” and “removals” of nodes over time

Does the Scheme Meet the Requirements?

- IoT System Operational Requirements (Empirical)
  - Authentication & Transaction integrity
    - Nodes and transactions are authenticated using the group key and the node Lamport signatures
    - A node uses his Lamport public key to validate inserted DATA, transmits DATA to aggregator(s)
  - Lightweight operations in terms of resources
    - Operations can be lightweight for sensors. Aggregators have more resources
  - Encryption is a plus but not firm requirement
    - No need for encryption

Does the Scheme Meet the Requirements?

- IoT System Operational Requirements (Empirical)
  - Capable of handling sensor “sleep/power-off” periods
    - Nodes can re-authenticate using their knowledge of historical transactions proving their membership specific historical transactions using **predecessors** for Lamport Signatures

$$T(x_i) = \underbrace{Data}_{\text{Transactional Data}} \parallel \underbrace{h\left(Data \| h^k(x_i^{k_0})\right)}_{\text{Data Signature}} \parallel \underbrace{h^{k-1}(x_i^{k_0}) \| \, x_i^{k_0}}_{\text{Signature Verification}} \text{ where } x_i^{k_0} \text{ is the key } k_0 \text{ for node } x_i$$

  - Handle resource diversity and data of sensors and aggregators
    - Different nodes store different portions of the ledger
    - Aggregators fully, others partial

Talk Outline

- Overview of IoT
- Security Failures in IoT: Motivating Use Cases
- **Why direct use of Blockchain is not practical for IoT**
- **Challenge**: Design practical Blockchain-based protocols for IoT
- Conclusions, Discussion & Challenges

Conclusions

- IoT Scale, Vendors, Technologies increase exponentially
- IoT Devices will always have diverse capabilities & Resources
- Use of Cryptography is done without clear understanding of the implications
- No Current Standards for Lightweight cryptography
- Blockchain inspired protocols combined with new Cryptographic primitives might be the path forward

Discussion

Now that we build a Blockchain for IoT what is next?

- Secure Software Updates and Transactional Cross-IoT
- Audit & Monitor Devices from different Vendors
- Enable Application Markets for IoT
- Share information using Blockchain Smart Contracts
- Verified Time for IoT

Are we Done? Challenges

Blockchain Technology

- Cost of Deployment & Energy is an open problem for IoT devices, Consumer products
- Bi-directonality of communications
  Scaling latency
  No msec or nsec transactions
  Time Verification
- Scalability & Interoperability not initial design tenets
  Communication Overhead
- Privacy & Security is not just immutability
  What about data provenance and removal?
  Blockchain is forever
- Novel Blockchain-inspired designs that adhere to requirements of the use cases
- Competing technologies are causing confusion and do not offer complete solutions for user needs
- Lack of Standards and maturity of technologies an impediment for adoption

Thank you, Questions?

Operational Transactions

$$T(x_i) = \underbrace{Operation \,||I_A\, ||\, x_n^{k_0}}_{\text{Administrative Data}} ||\ \underbrace{h\Big(Data||\, K_G||\, h^k(x_i^{k_0})\Big)}_{\text{Operation Signature}}\ ||\ \underbrace{h^{k-1}(x_i^{k_0})||\, k_0}_{\text{Signature Verification}}$$

where $Operation = \{ADD\ or\ REMOVE\}$ and $x_n^{k_0}$ is the node id (here node n) the operation is applied to. $I_A \in \{0, 1\}$ denotes if the added or removed node is an aggregator. We assume that node x_i broadcasted the transaction $T(x_i)$. In case of ADD operation $x_n^{k_0}$ denotes the first key of the newly added node n.