Secure Data Sharing and Searching at the Edge of Cloud-Assisted Internet of Things

Paper Authors: Muhammad Baqer Mollah and Md. Abul Kalam Azad, Athanasios Vasilakos
IEEE Cloud Computing March 2017

Presented by: Samantha Orogvany-Charpentier
Introduction

• By 2020, estimated 50 billion IoT devices will be connected to the internet
• IoT devices are part of larger platform
  • Large amount of data generated/Large Storage requirement
  • High computational requirement for processing and analysis
  • Secure and efficient requirement
• Cloud Platform provides
  • Virtually unlimited storage
  • Scalable processing
  • On-demand availability
• IoT requires low latency, high data rate, fast data access, real-time data analytics/processing. Cloud alone cannot support this. Edge computing helps.
**Edge Computing**

- **Edge Servers**
  - Examples: Personal mobile devices, network devices hosted within one hop, stand-alone servers
  - Provides data processing, communication, storage close to smart devices while also connected with cloud servers

- **Data Sharing for Smart Devices**
  - Data sharing between smart devices is essential component of IoT
  - Sharing at edge instead of centralized cloud model creates faster data access, higher bandwidth, lower latency
  - This creates potential security issues
    - Risks: Data leakage, data modification, integrity, unauthorized access
    - Essentials: Confidentiality, integrity, access control
  - IoT devices cannot handle typical computation-intensive operations for security
Proposed Lightweight Cryptographic Scheme

1. Create secure data-sharing scheme that uses both secret key encryption and public key encryption. Edge servers handle all security operations.

2. Create searching scheme for authorized users to search for desired data stored on edge/cloud.

3. Create verification process for shared data and data retrieval after searching (proving data integrity).

4. Analyze performance of scheme to show efficiency and efficacy for IoT use.
Cryptographic Mechanisms

Encryption Concepts
• Secret Key Encryption
• Public Key Encryption
• Searchable Secret Key Encryption
  • Searching specific data on outsourced storage encrypted data via a generated trapdoor. The data owner device needs to share the secret key with all authorized devices to generate the trapdoor.
• One-way Hash Algorithms
  • Verify data has not been modified with integrity checking via hash functions
• Digital Signature
  • Key pair, digital cert ensures identity of user or entity
System Architecture

- **Smart Devices**
- **Edge Servers**
  - Semi-trusted, secure.
  - Data storage and processing for smart devices.
  - Secret key generation/management
  - Encryption/decryption
- **Certificate Authority**
  - Fully trusted, issues certs to edge servers
- **Key Generation Servers**
  - 3rd party that generates key pairs

*FIGURE 1. Cloud-assisted Internet of Things scenario.*
Models and Assumptions

Threat Models
- Insider Threats
  - Initiated by malicious insiders (smart devices / edge servers) that want to access / disclose / modified stored/shared data
- Outsider Threats
  - Initiated by unauthorized outsider smart devices to access data

Trust Model and Assumptions
- Assume smart devices cannot encrypt, decrypt, generate keys themselves
- Assume edge servers are semi-trusted, can achieve security over shared/searched data
- Assume edge servers have Secret Key generator that delivers keys to other edge servers
1. Key Generation

Edge Server generate two types of private keys on behalf of smart devices (differently and uniquely)

- 256 bit keys randomly generated

1. Sec.Key - Data Sharing
2. S.Sec.Key - Searching

FIGURE 2. Proposed secure data-sharing scheme.
2. Data and Keyword Uploading

1. Edge Server (ES) stores username/password for data owner
2. IoT data is transferred to nearby ES
3. IoT sends keywords of data to ES for search
4. Key generation Server sends pair of keys to ES
5. ES encrypts data and keywords before uploading to cloud
6. CA issues digital certificate to verify ES
7. To ensure integrity, compute hash and sign hash with private key
   - $H_1 \leftarrow \text{Compute hash (Data)}$
   - $\text{Signed.}H_1 \leftarrow \text{Sign (}H_1, \text{ Private.Key)}$

ES uploads tuple to a table under *username* (Encrypted data, encrypted key, signed hash, sig)

(C.Share || C.Sec.Key || C.KW.Search || Signed.H1 || Dig.Cert)

**FIGURE 2.** Proposed secure data-sharing scheme.
3. Data Sharing and Downloading

1. Access Data: authorized IoT requests data from ES with username/password
2. ES downloads tuple under username
   1. (Encrypted data, encrypted key, signed hash, sig)
   2. (C.Share || C.Sec.Key || C.KW.Search || Signed.H1 || Dig.Cert)
3. ES checks digital cert
4. Decrypts private key (fails if not authorized)
5. ES decrypts data
   • Data ← Decrypt (C.Share, Sec.Key)
6. Verifies data with hash function
   • H2 ← Calculate hash (Data)
   • H1 ← Decrypt (Signed.H1, Public.Key)
   • Check (H1=H2)
7. Finally data is sent to authorized recipient

FIGURE 2. Proposed secure data-sharing scheme.
4. Data Searching and Retrieval

1. To search for data, auth user sends keyword to ES after login
2. ES receives secret key to generate trapdoor
3. Trapdoor sent to storage server with request to search
4. Storage server searches encrypted keywords under username, sends tuple back to edge server on success
5. ES checks digital certificate
6. Decrypts secret key and then decrypts to get data
7. ES verifies with hash function, sending data to device on success

**FIGURE 3.** Proposed secure data-searching scheme.
Performance for Edge Server

- Tested using Windows 7 32 bit AMD CPU 1.65 GHz and 2 GB RAM
- AES, RSA, SHA-256 for secret key, public key encryption, hash function
- Secret Key generation is 1.4 ms

FIGURE 4. Processing time of encryption and decryption with Advanced Encryption Standard.
Improvements and Conclusions

• Other researchers propose cert-less schemes
• Drawbacks are too much processing times due to relying on the cloud
• Authors plan to expand on authenticating and access control challenges
References

Bringing IoT and Cloud Computing towards Pervasive Healthcare

Paper Authors: Charalampos Doukas and Ilias Maglogiannis

By Danielle Aring
Background: Pervasive Computing and IoT in Health Care

• What IS pervasive computing?
  • Embedding computational ability into everyday objects!
  • Object can effectively communicate and perform tasks.
  • This minimizes end user’s interaction with computers as COMPUTERS.

• Pervasive healthcare apps utilizing body sensor networks (wireless networks of wearable computing devices)
  • generate massive data
  • stored and processed, for future use

• Cloud computing among with IoT concept is new trend for efficient processing and managing of sensor data online

• Paper presents a secure platform based on Cloud computing for management of mobile and wearable healthcare sensors, demonstrating the way the IoT paradigm applied on pervasive healthcare
Paradigm of Pervasive Healthcare

• Introduced awareness for
  • Independent living of elderly with disabilities
  • Consistent medical supervision of patients with chronic medical conditions in remote underserved, and isolated locations

• This enables a need for healthcare services delivered through a network on-demand, anywhere, at all times to any persons
Challenges: Pervasive Health Management Through Mobile Devices

- Data storage
- Data management
  - Physical storage issues
  - Availability
  - Maintenance
- Interoperability
- Availability of different resources
- Security and privacy
  - Permission control data anonymity
- Unified and ever-present access
Solution: Cloud Computing Pervasive Healthcare and IoT

- Introduce cloud computing framework to electronic healthcare system by using a network of remote servers hosted on the internet, to store manage and process data, rather than local servers.

- Cloud Benefits:
  - Access to shared resources
  - Common infrastructure in an ever-present and transparent way
  - On-demand services over a network
  - Operations performed to meet altered needs

- Advances in M2m allow direct interaction with pervasive healthcare sensors with the internet and by ext. Cloud computing (e.g IoT)
Benefits of IoT in Pervasive Healthcare

- Real time access to device information
- Remote management of devices
- Scalable/globally accessible web applications
- communication interfaces to external applications.
Cloud Computing Idea
Problem

- Number of cloud computing platforms for pervasive management of user data exist:
  - ICloud, Pithos, Dropbox
  - Commercial (Amazon AWS, Rackspace)
- Most do NOT provide developer support to create custom APPS
  - And incorporate Cloud computing functionality (only AWS)
  - NONE are optimized for the provision of services to health-care based applications
- Health Care data is often highly sensitive so the system must be secure
  - Generally involves authentication and cryptography
System Architecture

1. Wearable/mobile Sensors (acquire data)
2. Sensor Gateway (collects signals from sensors)
3. Communication APIs (REST web Services)~>cloud provided used by sensor gateway
4. Managing Application (web-based visualization sensor data)
5. Cloud Infrastructure (hosts interfaces and managing app) (essential resources CPU storage)
Architectural Features

- Scalable- (relies on Cloud infrastructure provides resources based on utilization and demand)
  - More users sensors and other data sources can be added without loss to system functionality, maintenance or expansion
- Interoperability- (w/ (external applications) due to web-services base interfaces )
- lightweight access - (REST) lightweight, easy access implemented by wireless sensor and mobile platforms.