



Efficient and secure searchable encryption protocol for cloud-based Internet of Things

Libing Wu^a, Biwen Chen^a, Kim-Kwang Raymond Choo^{b,c}, Debiao He^{a,d,*}

^a State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China

^b Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA

^c School of Information Technology & Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia

^d Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China

HIGHLIGHTS

- We define a searchable encryption model achieving efficiency and security for Cloud of Things.
- We propose an efficient and secure searchable encryption protocol for Cloud of Things.
- We show that our proposed protocol is provably secure. It satisfies the security requirements: inside KGA resilience forward privacy and file-injection attack resilience.
- Detailed performance analysis and experimental result are given.

ARTICLE INFO

Article history:

Received 29 March 2017

Received in revised form 28 June 2017

Accepted 17 August 2017

Available online 24 August 2017

Keywords:

Internet of Things

Cloud-of-Things

Searchable encryption

Forward privacy

File-injection attack resilience

Insider keyword guessing attack resilience

ABSTRACT

Internet of things (IoT) applications comprising thousands or millions of intelligent devices or things is fast becoming a norm in our inter-connected world, and the significant amount of data generated from IoT applications is often stored in the cloud. However, searching encrypted data (i.e. Searchable Encryption—SE) in the cloud remains an ongoing challenge. Existing SE protocols include searchable symmetric encryption (SSE) and public-key encryption with keyword search (PEKS). Limitations of SSE include complex and expensive key management and distribution, while PEKS suffer from inefficiency and are vulnerable to insider keyword guessing attacks (KGA). Besides, most protocols are insecure against file-injection attacks carried out by a malicious server. Thus, in this paper, we propose an efficient and secure searchable encryption protocol using the trapdoor permutation function (TPF). The protocol is designed for cloud-based IoT (also referred to as Cloud of Things – CoT) deployment, such as Cloud of Battlefield Things and Cloud of Military Things. Compared with other existing SE protocols, our proposed SE protocol incurs lower computation cost at the expense of a slightly higher storage cost (which is less of an issue, considering the decreasing costs of storage). We also prove that our protocol achieves inside KGA resilience, forward privacy, and file-injection attack resilience.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

In an Internet of Things (IoT) architecture, there are many real-world objects (also referred to as devices or things) connected to the Internet. These interconnected objects (e.g. sensors, mobile devices such as Android and iOS devices, wearable devices, and drones or unmanned aerial vehicles) are responsible for sensing, collecting, disseminating and exchanging data in a broad range of context, such as public/homeland security (e.g. smart cities),

* Corresponding author at: State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China.

E-mail address: hedebiao@163.com (D. He).

utility (e.g. smart grids), logistics (e.g. smart supply chains), and intelligent building (e.g. smart homes). The trend of IoT in our modern society is explained in a recent report from Gartner, which estimated that 63 million IoT devices will be attempting to connect to the network each second by 2020 [33].

Cloud computing can also play a supporting role in IoT architecture, as explained by Roopaei, Rad and Choo [28]. The authors used the Cloud of Things-based automated irrigation as a use case to illustrate how integrating cloud and IoT can make energy use more efficient and less costly. This is not surprising, considering that cloud computing can provide affordable and real-time computing and storage capabilities [30].

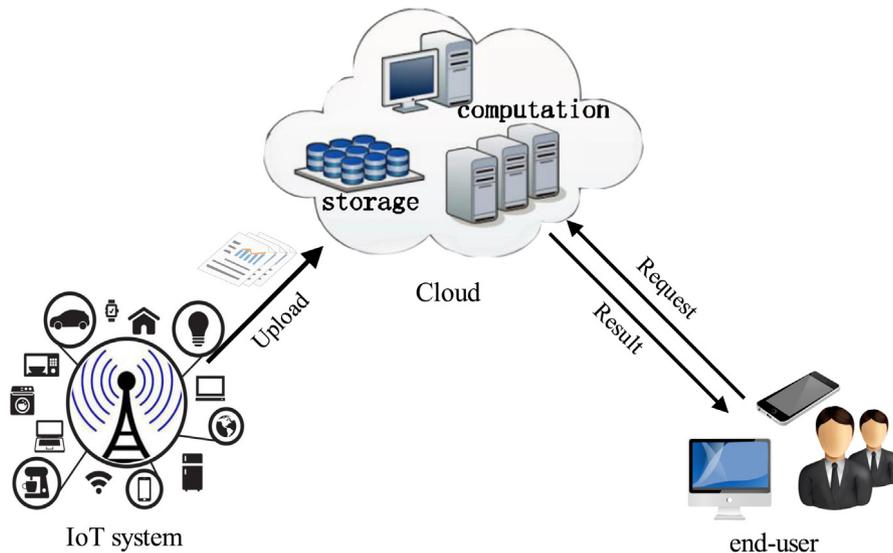


Fig. 1. A typical Cloud of Things (CoT) system.

A typical cloud-based IoT system is shown in Fig. 1. In a CoT environment, data collected by various smart devices is uploaded to the cloud server. Then, the user obtain information of interest from the cloud server via a client device. Since it is not realistic to expect that a cloud server is completely trustworthy (e.g. the server may be compromised and there may exist a corrupted or malicious insider) [14,18], data outsourced to or stored in the cloud should be protected (e.g. using a secure encryption scheme). However, searching on encrypted data is challenging given today's technology. Thus, Searchable Encryption (SE) has emerged as a salient research inquiry. A SE protocol is designed to allow one to search on encrypted data containing specified keywords and to obtain the response from the server based on the keyword trapdoor without the need to decrypt the data. The server is also prevented from learning the content of the user's query.

A number of SE protocols/schemes have been proposed in the literature since the work of Song et al. [29], and these SE protocols can be broadly categorized into searchable symmetric encryption (SSE) protocols (see [6,9,15,19,23,29,31]) and public-key encryption with keyword search (PEKS) protocols (see [1–3,5,12,16,17,24,34–36]). It is known that the SSE protocols generally are more efficient, but suffer from complex and expensive key management distribution [10] limitations (due to the fact that data owner needs to share a key with each user by the secure channel [4]). We refer the interested reader to a recent review of SSE schemes by Poh et al. [26].

PEKS protocols are known for their stronger security and flexibility, but a key limitation with such protocols is the inability to resist inside keyword guessing attacks [7] (e.g. from a malicious server or cloud employee). Specifically, the cloud server can use the user's public key to encrypt some keywords and use the keyword's ciphertext to test the content of a trapdoor. Recently in 2016, Chen et al. [13] proposed a PEKS protocol to resist such an attack, but their protocol is insecure against an external adversary.

We also observe that most SE protocols are vulnerable to file-injection attacks and have weak forward privacy. For example, Zhang et al. [37] demonstrated that in a file-injection attack, an adversary can recovery all of the user's trapdoors using very few injected files. Such an attack is clearly a threat to SE protocols, undermining the privacy of user data. In summary, if one want to deploy existing SE protocols in a CoT environment, the following limitations need to be addressed:

1. Security requirements: As evidenced by the findings reported in [13,37], there is a need to design protocols that are secure

against inside keyword guessing attacks and file-injection attacks in addition to achieving other standard security properties.

2. Computational overheads: The amount of data generated by CoT devices in some applications may be significant (i.e. big data issues). Thus, there is a need to design efficient protocols with low computational overheads that are suitable for CoT deployment.

Therefore, in this paper, we construct an efficient and secure SE protocol using the trapdoor permutation function. The protocol uses neither bilinear pairing operation nor map-to-point hash operation. The search time of protocol is only related to the database update times. We then demonstrate the security and evaluate the performance of the proposed protocol.

We will briefly review related literature in the next section, before presenting the relevant background materials in Section 3. We present the proposed protocol in Section 4. In Sections 5 and 6, we analyze the security and evaluate the performance of our protocol, respectively. Finally, we conclude the paper in the last section.

2. Related literature

In this section, we present the related existing SE protocols and roughly categorize them according to their design goals, namely: efficiency, application and security.

Efficiency: The search time is one of the key factors in determining the efficiency of SE protocol. Song's [29] search time is linear to the database size, thus the protocol is inefficient in a big data environment. To mitigate such a limitation, Curtmola et al. [15] presented an index-based SSE construction to achieve sublinear search time. The complexity of the index-based protocol is related to the keyword space. In 2013, Cash et al. [9] presented a highly-scalable SSE, designed to support very large database. From existing literature, a practical SE protocol should minimize the search time, which is not a surprising observation.

However, in most PEKS protocol, complex operations (e.g. bilinear pairing, map-to-point hash) affect efficiency. To reduce the computational complexity, Di [16] proposed a PEKS protocol without bilinear pairing operation. However, the protocol is not practical. Recently, using homomorphic smooth projective hash functions, Chen [13] designed a protocol that does not require the bilinear pairing operation. Thus, their protocol is more efficient. In other words, the design of a SE protocol should avoid having complex operations.

Download English Version:

<https://daneshyari.com/en/article/4951512>

Download Persian Version:

<https://daneshyari.com/article/4951512>

[Daneshyari.com](https://daneshyari.com)