Attribute Based Content Security and Caching in Information Centric IoT

Anonymous

ABSTRACT

Information-centric networking (ICN) is a Future Internet paradigm which uses named information (data objects) instead of host-based end-to-end communications. In-network caching is a key pillar of ICN. Basically, data objects are cached in ICN routers and retrieved from these network elements upon availability when they are requested. It is a particularly promising networking approach due to the expected benefits of data dissemination efficiency, reduced delay and improved robustness for challenging communication scenarios in IoT domain. From the security perspective, ICN concentrates on securing data objects instead of ensuring the security of end-to-end communication link. However, it inherently involves the security challenge of access control for content. Thus, an efficient access control mechanism is crucial to provide secure information dissemination and access. In this work, we investigate Attribute Based Encryption (ABE) as an access control apparatus for information-centric IoT. Moreover, we elaborate on how such a system performs for different parameter settings such as different numbers of attributes and file sizes.

CCS CONCEPTS

• **Security and privacy** → **Access control**; *Cryptography*; *Distributed systems security*; • **Networks** → **Security protocols**;

KEYWORDS

IoT, Information-centric networks, Ciphertext policy attribute-based encryption, Secure communications

ACM Reference Format:

Anonymous. 2018. Attribute Based Content Security and Caching in Information Centric IoT. In *Proceedings of Under review (Submitted to ARES '18 (under review))*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Information Centric Networking (ICN) is one of the most promising approaches for Future Internet since estimations show that video traffic will reach 79% of the Internet traffic by 2018 [5]. In ICN, content owner publishes the contents which are stored in caches. When a host wants to reach a content, it fetches the content directly from a network-resident cache instead of communicating with the content owner. Therefore, reduced congestion and less traffic load with higher network performance are possible. In a similar vein, content fetching is a key operation in IoT. It is typical that same data can be requested more than once in IoT environment. Therefore, caching is instrumental to eliminate extra transmissions and increase resource efficiency. Moreover, improved robustness and dissemination are important for IoT communications. In that regard, ICN can alleviate various challenges ranging from energy efficiency to latency requirements for IoT communications.

Unlike host-centric security, ICN requires location independent security mechanisms to enable ubiquitous in-network caching. Endpoint security is not sufficient since contents are also supposed to be secure themselves. Therefore, security model for ICN should provide an information oriented data integrity and authenticity check mechanism. There are also ICN specific threats like content poisoning and cache pollution attacks [15]. While the goal of content poisoning attack is to fill the router caches with fake contents, the goal of cache pollution attack is to degrade cache effectiveness and increase the content retrieval latency. With the integration of ICN paradigm into IoT, these security challenges are aggravated with the pecularities of IoT such as device heterogeneity, resource constraints, lack of central management and diverse application requirements [13]. First of all, the owner of a content is not responsible for the access control because he does not have control over distributed caches in the network. Thus, there should be an access control policy which can securely drive content consumption while being efficient and scalable for IoT nodes. To address these security problems, Attribute Based Encryption (ABE) is instrumental since it allows several different groups of users by defining attributes rather than relying on the identities [11].

ABE is a variant of public key encryption that does not only use public-private key pair but also a well-defined access policy to encrypt the plaintext by providing fine grained access control on the encrypted data [14]. However, it also reveals some important issues related to management, cost and security. In information-centric networks, objects are cached in different locations and accessible when requested by content consumers. Due to the distributed nature of the system, attribute management is difficult from the scalability perspective. Additionally, ABE induced processing is supposed to be optimized avoiding a significant penalty on IoT device performance.

ABE has two different implementations, namely Ciphertext Policy Attribute Based Encryption (CP-ABE) [3] and Key Policy Attribute Based Encryption (KP-ABE) [7]. In CP-ABE, the access policy is associated with the ciphertext and entities with the correct attributes are able to obtain the plaintext. In KP-ABE, the access policy for decrypting the ciphertext is encoded into secret keys of entities and ciphertext is generated by using a set of attributes. Entities with the correct access policy are able to decrypt the ciphertext. Although the main purpose of KP-ABE and CP-ABE is to provide access control for file sharing systems, CP-ABE schemes have various application areas such as sharing content in ICN [11],

Submitted to ARES '18 (under review), 2018, Hamburg DE

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

secure communications among IoT devices [16], and security of personal health record [6].

In this work, we devise a content security approach via CP-ABE which jointly works with attribute-based cache management for information-centric IoT. The key objective of this scheme is the provision of access control while opportunistically using the attribute information to improve caching performance. In addition to that, we investigate the system performance for different parameter settings such as different number of attributes and varying file sizes. The rest of this paper is organized as follows. Section 2 presents the related works in the literature. Section 3 describes the attribute based content management and caching for ICN in IoT. Next, we elaborate on the security analysis of the investigated system. Finally, we present the experimental results and discuss the performance of the proposed solution in Section 5 and conclude this paper in Section 6.

2 RELATED WORK

Efficient delivery of content without compromising the data security is one of the key requirements of IoT. This is also valid for information-centric operation in that environment. There are various works in the literature elaborating on different aspects of ABE and ICN. In [10], Li et al. use a modified version of ABE for ICN naming scheme to preserve privacy in access control and provide flexible attribute management. In their ontology based attribute management approach, the content owner pushes his data by specifying which attributes are used for access control and which ones are used for content search. Therefore, content names are protected based on attributes and attribute combination operations in the access control policies are supported with reduced cost. Similarly in [9], Ion et al. use combination of ICN names with Boolean expressions of constraints on attributes in order to enable users to express more complex requests.

Another related research challenge is the computational complexity of ABE. This characteristic is extensively investigated in IoT environment since ABE requires computationally-expensive operations which may be a challenge for resource constrained devices. In that regard, a typical approach is to perform ABE operations on a gateway or servers which are computationally more powerful than IoT devices. However, such an architecture requires secure communication and trust between the device(s) and the gateway or the server. In [4], ABE is carried out on sensors to encrypt the symmetric key with a predefined access policy. The results show that it is feasible on resource constrained devices depending on the use case and the security requirements. Similarly in [11], ABE is used at sensor nodes before pushing sensory data to the ICN environment. The authors point out that ABE can be implemented for devices that have low memory and battery power although it is a processing intensive task. Apparently, the significant factor is the choice of right level of security and the number of employed levels.

The conventional mobile user devices have also been investigated in the literature. In [2], ABE has been applied to smart phone devices by considering some constraints such as battery and memory. Similarly, a Java-based application of ABE on Android smart phones has been implemented in [17]. Another IoT focused paper by Oualha et al. describes a new technique that was applied on ABE algorithm to overcome the computational load for resource constrained devices [12].

3 ATTRIBUTE BASED CONTENT SECURITY AND CACHING (AB-CSC) FOR INFORMATION-CENTRIC IOT

The peculiar characteristics of information-centric IoT calls for novel techniques utilizing content-centric paradigm for improving security. In this section, we introduce AB-CSC. First, we give the overall organization of the proposed approach. We focus on an architecture where some IoT devices act as content aggregators while some others consume them. Then, we describe the content management and caching mechanism of AB-CSC. Finally, we present the role of CP-ABE in AB-CSC.

3.1 Overview of AB-CSC

AB-CSC is a content caching mechanism that operates on ICNs consisting of IoT devices. Components of AB-CSC can be defined as follows:

- **Producers:** In AB-CSC environment, each IoT device is defined as a producer in the network. The sensitive information collected by sensors of IoT devices is encrypted and transmitted to content repositories (caches).
- **Content Repositories:** This is the intermediary set of nodes responsible for caching the encrypted content generated by producers.
- **Consumers:** Entities that request the sensitive information from content storages. These devices process and utilize fetched data according to various use-cases.

An illustration of AB-CSC operation is as shown in Figure 1. Generated data (e.g. from environmental sensors in a smart city scenario) are collected through producers and transmitted to cache storages using the closest relay in the network in an encrypted manner. Encryption is realized via using CP-ABE by using a predefined set of access policy based on the attributes of consumers. Therefore, producer and consumer relation is satisfied before the deployment of an IoT device. Cache selection process is executed based on caching attributes of content. The encrypted sensory data are put into the selected cache ((1) in Figure 1). In order to get the content from cache storages, consumers make the request via using the closest relay in the network. Then, the interest is forwarded to the cache that the desired content is cached ((2) in Figure 1). When the content is fetched, consumers use the set of attributes to decrypt the content ((3) in Figure 1). If a consumer does not have the correct set of attributes, this consumer can not access the content. Therefore, end-to-end security for content delivery is achieved.

3.2 Cache Management and Selection in AB-CSC

The cache related mechanisms in AB-CSC are comprised of two main schemes, namely cache management and cache selection. The former refers to the algorithm for cache placement and eviction according to content requests in each cache while the latter refers to the selection of caches in the network used for keeping contents according to content attributes.

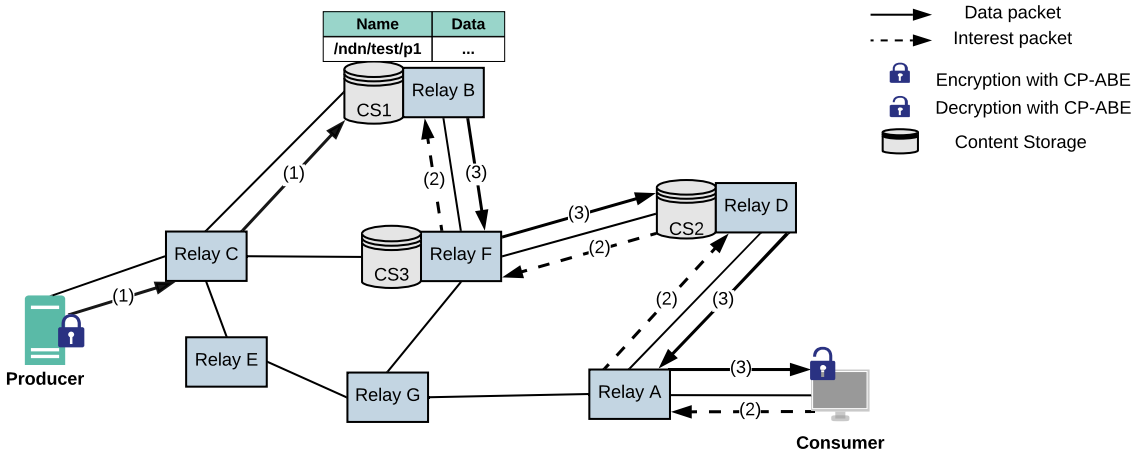


Figure 1: An illustration of AB-CSC operation in information-centric IoT environment.

Algorithm 1 Cache Management

Input:

x_{cs}: item x of cache cs
x_{nc}: item x of new content nc
s_e: size of entity e
p_c: popularity of content c
i_c: cache affinity of content c
r_c: criticality level of content c
f_c: sampling frequency of producer of content c

1: **procedure** CacheManager
2: $i_{nc} \leftarrow p_{nc} + 1/f_{nc} + r_{nc}$
3: **while** $s_{cs} + s_{nc} >$ maxCacheSize **do**
4: delete c which has min i_c in cs
5: push nc in cs

3.2.1 Cache Management Scheme. Cache management is a critical determinant on any information-centric network architecture [8]. The need for a cache management mechanism becomes a significant issue due to the resource limitations of IoT devices [18] such as overloading the communication bandwidth with frequent content requests and fast battery depletion problems. Therefore, we propose a cache management heuristic by considering the popularity, criticality and sampling frequency of contents. We determine the caching index, i.e. the cache affinity for a content c, based on the following criteria:

- **Popularity of the content** p_c: Content popularity is one of the most important criteria for the cache replacement because consumers tend to request popular contents more frequently. To increase cache hit ratio in our system, we keep popular contents in a cache more likely, compared to unpopular contents.
- **Sampling frequency of the producer sensor** f_c**:** We use sampling frequency as a secondary criterion which may be critical for our context. For example, when the latest value recorded by a sensor is requested, the returned data may be outdated because the corresponding values have been updated. Therefore, the content of a sensor which has low sampling frequency will be kept longer in a cache than a sensor which has high sampling frequency.
- **Content criticality** r_c**:** We classify generated contents into C criticality groups. Although there is a general correlation between criticality and frequency, there are also event-driven use cases for an IoT system. A typical example is the surveillance of a critical infrastructure. When there is a physical intrusion alert, the sensor data (e.g. video data from motion-sensing cameras) are very critical despite not being frequently generated.

With these three criteria, we calculate caching index (affinity) i_c of a content. This metric determines the of a content To ensure criteria effect to i_c equally, we normalize each criterion (p_c, f_c, and r_c) to the range [0,1] using the maximum values. Moreover, the time-to-live (TTL) of contents are predefined to represent how long it takes before a content becomes stale and loses its utility in the system. Any content which exceeds its TTL period is removed from the respective cache. The cache management algorithm of AB-CSC is given in Algorithm 1.

After a new content arrives to a cache, its cache index is calculated and attached to that content. Then, if the cache has enough space to keep the new content, the content is stored in the cache. Otherwise, among the stored contents which is stored in the cache, the content(s) which has the minimum index is removed (evicted) until the cache has enough space for the incoming content.

3.2.2 Cache Selection Scheme. After a content is generated by an IoT node, it is critical to decide on which cache(s) to store the content. An efficient cache selection scheme is essential to reduce cache fetch distance. A smaller distance leads to reduced traffic and less overhead in the network. To achieve that goal, we utilize a cache selection scheme based on attributes of consumers.

Due to the ABE architecture, our consumers have attributes and also contents are encrypted with consumer attributes. To build

Table 1: Access Structures of Contents

Content name	Title	Department	Experience
Content 1	Technician or Engineer	Department A	Senior or Junior
Content 2	Engineer	Department B	Senior
Content 3	Team Leader	Department B or Department C	Senior or Junior

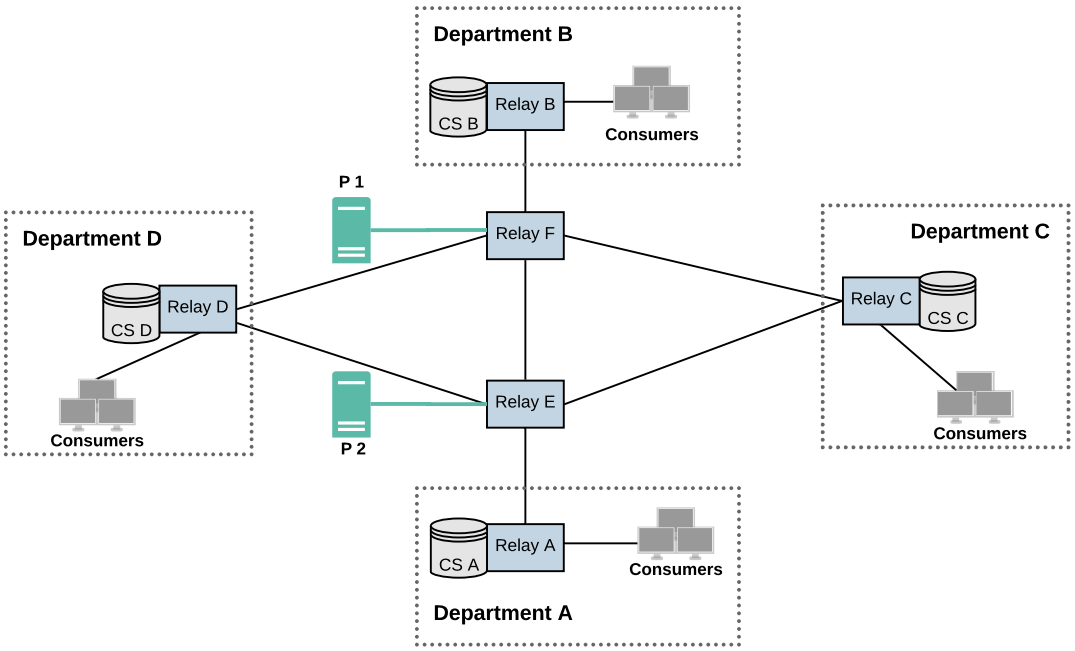


Figure 2: An illustration of a sample company topology.

our cache selection scheme, we take advantage of consumer attributes. In the system setup, we specify one of the attributes as a cache selection attribute. Our cache selection attribute should indicate geographically closeness between consumers who have the attribute. Then, we store contents which have the same cache selection attribute at the same cache.

To provide further understanding, we explain our cache selection scheme on a toy example. An illustration of a company topology is shown in Fig. 2. Produced contents and their access structures are as given in Table 1. Assume that Content 1 is encrypted under following access structure: $(Technician \vee Engineer) \wedge DepartmentA \wedge (Senior \vee Junior)$. In a system setup, since department attribute indicates geographically closeness between consumers, we choose department attribute as a cache selection attribute. To select a cache for Content 1, our cache selection scheme decide which cache is more suitable for Content 1 by using the cache selection attribute. In this example, the cache of Department A is selected for Content 1. Since Content 1 is encrypted by using the access policy that covers the Department A attribute, it can be only accessible by the members of Department A. Therefore, AB-CSC stores Content 1 in the closest cache to the Department A. By using the same approach, Content 2 is stored in Cache B and Content 3 is stored in both Cache B and Cache C.

3.3 Content Encryption/Decryption using CP-ABE

Content encryption and decryption operations of AB-CSC are realized by using CP-ABE scheme in [3]. In AB-CSC, there exists a direct relationship with consumers and their producers. Each producer encrypts the content to be shared by using CP-ABE and the access policy that is related to attributes of consumers. Then, consumers can decrypt the shared content via using their attributes and secret keys. Consumers that do not have the correct attributes are not able to decrypt the content.

In AB-CSC, we assume that the following conditions are exist. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_n$ be the set of IoT devices that collects data to be used by the set of consumers $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ and $\mathcal{A} = A_1, A_2, \ldots, A_m$ be the set of attribute sets for consumers, where

A_j is the set of attributes of C_j. Mapping between $\mathcal{S}$ and C is established by using a relationship matrix denoted as $M_{n\times m}$, where rows of the binary matrix correspond to the set of consumers $\mathcal{S}$, columns of the matrix correspond to the set of consumers C. If $M_{i,j} = 1$ then the device S_i produces content for consumer C_j and $M_{i,j} = 0$ if there is no producer-consumer relation between S_i and C_j. We assume that access policies and public keys of each consumer in C are distributed to IoT devices before deployment. We also assume that the set of master keys $\mathcal{MK} = \{MK_1, MK_2, \ldots, MK_n\}$, where MK_i is the master key of S_i, are distributed to both IoT devices and related consumers. Then, CP-ABE of AB-CSC operates as shown in the Algorithm 2.

Algorithm 2 Content Encryption/Decryption using CP-ABE

1: **for** each source $S_i \in \mathcal{S}$ **do**
2: Get list of consumers $C_i' = \{\forall i, C_i | M_{i,j} = 0\}$
3: $Setup(\lambda, \mathcal{A}')$ S_i uses public parameters of consumers in C_i' and the master key MK_i by using the security parameter λ and the subset of attributes $\mathcal{A}' \subseteq \mathcal{A}$ of consumers in C_i'.
4: $Encrypt(PK, T_i, \mathbb{A}')$: S_i encrypts the content T_i, by using the access structure that consists of the attributes in $\mathbb{A}'$ and the public keys of C_i' to generate ciphertext CT_i.
5: $Key\ Generation(MK, \mathcal{A}')$: Each $C_j \in C_i'$ generates the secret key SK_j by using the master key MK and the set of attributes $\mathcal{A}'$.
6: $Decrypt(PK, CT_i, SK)$: Each $C_j \in C_i'$ decrypts the ciphertext CT_i by using the public key PK_j and the secret key SK_j. If SK associated with $\mathcal{A}'$ satisfies the access structure $\mathbb{A}'$ then, the C_j decrypts the message T_i.

AB-CSC also provides dynamic update operations for adding new producers into the network or adding/removing consumers to/from the network. Details of dynamic update operations are as follows:

- **Adding New Producers:** In AB-CSC, each producer is associated with the set of consumers. While adding a new producer into the network, the new device is also a producer for a set of consumers in the network. Let S_{n+1} be the new producer for the set of consumers $C' = \{C_1, \ldots, C_k\}$, where $k < m$. Then, the setup procedure of Content Encryption/Decryption algorithm is executed to obtain master key MK_{n+1} and to compute public parameters for consumers in C'. Moreover, access structure is also installed to S_{n+1} before the deployment.
- **Adding New Consumers:** Let C_{m+1} be the new consumer that will retrieve information from producers $\mathcal{S} = \{S_1, \ldots, S_t\}$, where $t < n$, after joined the network. Then, each producer $S_i \in \mathcal{S}'$ executes setup procedure of Content Encryption/Decryption algorithm is to obtain new master key MK_i and to compute new public parameters for consumers in $C' \cup C_{m+1}$, where C' is the set of consumers associated with the producer S_i. Moreover, access structure of C_j is also disabled for the further encryption operations. New consumers are not able to decrypt the shared content in cache storages since they do not have the necessary credentials to decrypt the content. Therefore, new consumers need to directly retrieve contents from producers.
- **Removing Consumers:** In AB-CSC, we also provide a procedure for leaving consumers. When a consumer C_j leaves the network, he/she cannot decrypt new contents encrypted by the set of associated producers. Let $\mathcal{S}' = \{S_1, \ldots, S_t\}$, where $t < n$ be the associated producers for the consumer C_j. Then, each producer $S_i \in \mathcal{S}'$ executes setup procedure of Content Encryption/Decryption algorithm is to obtain new master key MK_i and to compute new public parameters for consumers in $C' - C_j$, where C' is the set of consumers associated with the producer S_i. Moreover, access structure of C_j is also disabled for the further encryption operations.

In addition to the dynamic update operations, AB-CSC periodically updates attributes to provide secure content delivery from producers to consumers.

4 DISCUSSIONS ON THE SECURITY OF AB-CSC

In this section, we give discussions on the security of AB-CSC. The proposed approach uses CP-ABE as an access control mechanism for the content to be shared. As given in [3], the security of CP-ABE scheme can be considered by using the following cases:

- **Collusion Attacks:** Group of consumers combine their keys in order to decrypt the ciphertext that they could not decrypt individually [14].
- **Passive Attack:** An attcker is able to retrieve information from ciphertext by observing the communication.
- **Active Attacks:** A malicious entity in the network is able to intercept the communication and update the shared content by producers.

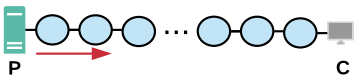
For the security against collusion attacks, as defined in [3], secret keys are randomized and they cannot be combined. In order to decrypt the ciphertext, an attacker has the predistributed key with the correct set of attributes. Therefore, CP-ABE is secure against collusion attacks.

For the security against passive attacks, an adversarial model was defined in [3]. According to this adversarial model, CP-ABE is secure against chosen-plaintext attacks. Adversary generates secret keys by using the sets of attributes $S_1, \ldots, S_{q_1}$ and the access structure A^*, where no $S_i \in \{S_1, \ldots, S_{q_1}\}$ satisfies A^*. Then, adversary sends two plaintexts M_1 and M_2, where $M_1 = M_2$, with A^* to the challenger. Challenger use A^* and randomly selects $b \in \{1, 2\}$ to encrypt M_b. The ciphertext is submitted to the adversary. Adversary repeats generating secret keys by using the sets of attributes $\{S_{q_1+1}, \ldots, S_q\}$ that does not satisfy the access policy A^* to correctly guess whether $M_b = M_1$ or $M_b = M_2$ by outputing the b' of b. Thus, as defined in [3], the advantage of adversary is $Pr[b' = b] - \frac{1}{2}$.

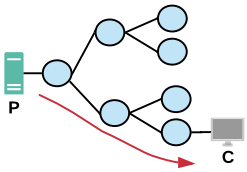
For the security against active attacks, as addressed in [3], the adversarial game can be extended to handle the chosen-ciphertext attacks.

5 EXPERIMENTAL RESULTS

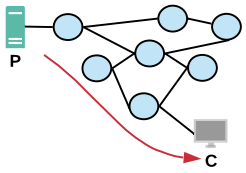
We have performed simulation based experiments to investigate the performance characteristics of AB-CSC. All simulations were



(a) Linear cascading topology



(b) Hierarchical topology



(c) Randomly generated topology

Figure 3: Topology types in the experiments. (P: producer, C: consumer)

carried out by using CCN-lite, which is an implementation of the Content Centric Networking protocol CCNx [1]. In order to perform our tests, we use linear cascading, binary tree and randomly generated topologies as shown in Figure 3 by using the system and simulation parameters listed in Table 2.

Table 2: System and Simulation Parameters

Parameter	Value
Average content size	1 kB
Total number of nodes	50
Total number of sensors	30
Average sensor frequency	1 content/10 minutes

First, we investigate the cache hit ratio by considering different maximum cache size values to see the effectiveness of our attribute based caching approach. As shown in Figure 4, we have simulated the average hit rate for popularity distributions by considering a varying Zipf exponent α. The cache size is described as a proportion of total content bytes generated in simulations and ranges from 1% to 10%. The cache hit ratio increases with the cache size as expected.

As shown in Figure 5, we have performed simulations to measure effect of criticality level on hit ratio. In these simulations, we have used three different criticality levels for contents, where the level 1 indicates the highest criticality and the level 3 indicates the lowest criticality. As shown in the simulation results, the hit ratio increases with the criticality of shared contents.

In order to show the applicability of AB-CSC, we have also made simulations to address the performance of CP-ABE in AB-CSC. First, we have measured the performance of CP-ABE regarding the effect of the number of attributes on total execution time of AB-CSC as shown in Figure 6. As expected, if the number of attributes

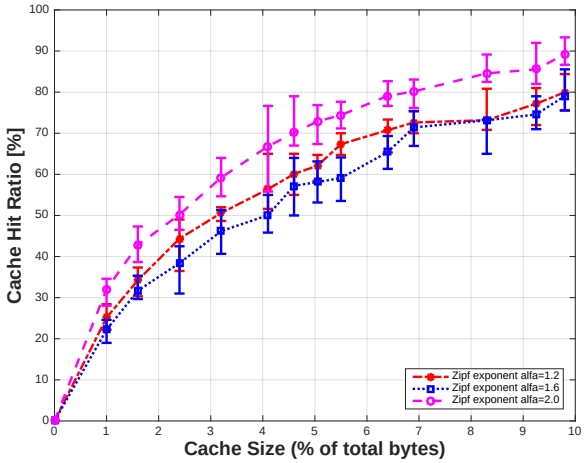


Figure 4: Average Hit Rate for Popularity Distributions (varying Zipf exponent)

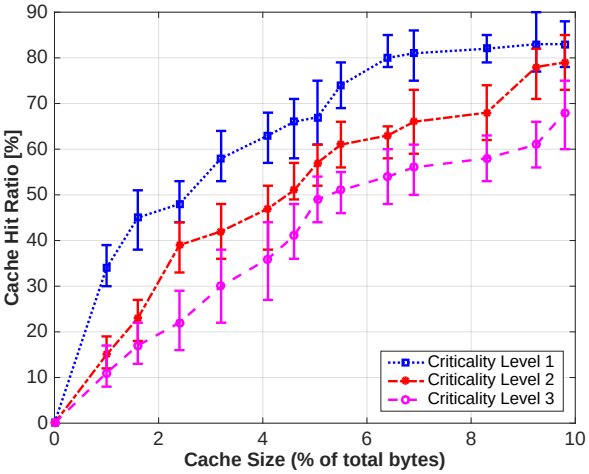


Figure 5: Simulations for Average Hit Rates by using Criticality.

used for encrypting the content increases then the total execution time for key generation and push encrypted content into cache also linearly increase. Time to retrieve and to decrypt requested content from cache also increases, because, using more attributes increases the size of a decrypted content. Thus, the increase in the number of attributes also increases the total execution time of AB-CSC. Such result can be seen as a burden for the overall execution time of the system. However, if more attributes are used for encrypting the content, then the accessibility of that content by different consumers also increases. Therefore, it is possible to use an aggregated access policy to share the content for aggregated set of consumers that are belong to different organizations.

Furthermore, we have also made simulations to evaluate the effect of the file size on the performance of encryption and decryption operations as shown in Figure 7. CP-ABE does not cause excessive

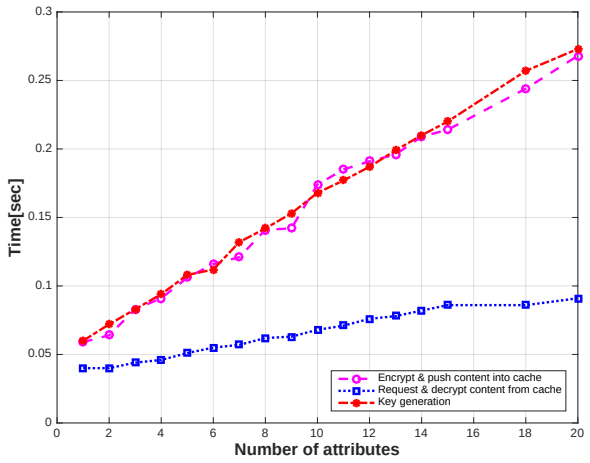


Figure 6: Effects of the Number of Attributes on the Execution Time of AB-CSC.

overheads in terms of the execution time of AB-CSC. Overhead of the execution time of CP-ABE in IoT environment can be considered as negligible since the produced content in IoT environment can be considered as relatively small.

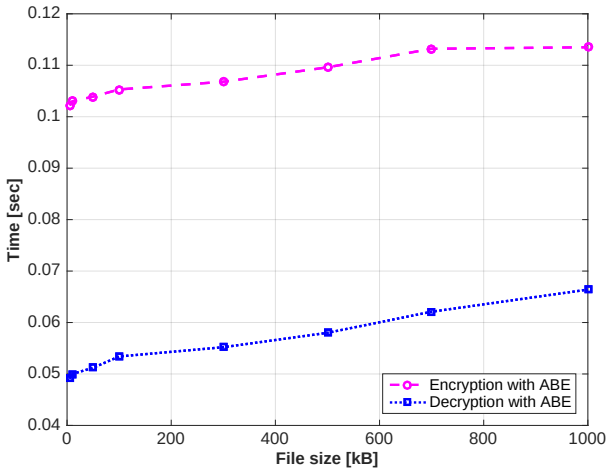


Figure 7: Effect of the Content Size on Decryption and Encryption Execution Time.

6 CONCLUSION AND FUTURE WORKS

In this study, we have proposed an Attribute-Based Content Security and Caching for information centric IoT, namely AB-CSC. The key contribution of this scheme is the provision of access control while opportunistically using the attribute information to improve caching performance. In addition, we have investigated the system performance via using simulations for analyzing the avarage hit rate by using different criteria of contents such as popularity distributions and criticality. Simulation results show that AB-CSC can be used as a secure cache selection mechanism. Moreover, we have simulated the applicability of CP-ABE as an access control mechanism in AB-CSC. According to the simulation results the increase in the number of attributes increases the total execution time of AB-CSC. However, such increase cannot be considered as a drawback for AB-CSC. If an access policy has more attributes than usual, then the shared content can be accessible by different consumers.

Implementation of AB-CSC in a real life application in order to analyze the energy consumption of IoT devices is left as a future work.

REFERENCES

[1] [n. d.]. CCN Lite: Lightweight implementation of the Content Centric Networking protocol. ([n. d.]). http://ccn-lite.net

[2] Moreno Ambrosin, Mauro Conti, and Tooska Dargahi. 2015. On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*. ACM, 49–54.

[3] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 321–334.

[4] Joakim Borgh, Edith Ngai, Börje Ohlman, and Adeel Mohammad Malik. 2017. Employing attribute-based encryption in systems with resource constrained devices in an information-centric networking context. In *Global Internet of Things Summit (GIoTS), 2017*. IEEE, 1–6.

[5] Cisco. 2016. White paper: Cisco VNI Forecast and Methodology, 2015-2020. (2016). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[6] Jieun Eom, Dong Hoon Lee, and Kwangsu Lee. 2016. Patient-Controlled Attribute-Based Encryption for Secure Electronic Health Records System. *Journal of Medical Systems* 40, 12 (06 Oct 2016), 253. https://doi.org/10.1007/s10916-016-0621-3

[7] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*. ACM, 89–98. https://doi.org/10.1145/1180405.1180418

[8] Gürkan Gür. 2015. Energy-aware cache management at the wireless network edge for information-centric operation. *Journal of Network and Computer Applications* 57 (2015), 33 – 42. https://doi.org/10.1016/j.jnca.2015.06.009

[9] Mihaela Ion, Jianqing Zhang, and Eve M Schooler. 2013. Toward content-centric privacy in ICN: Attribute-based encryption and routing. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 39–40.

[10] Bing Li, Ashwin Prabhu Verleker, Dijiang Huang, Zhijie Wang, and Yan Zhu. 2014. Attribute-based access control for ICN naming scheme. In *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 391–399.

[11] Adeel Mohammad Malik, Joakim Borgh, and Börje Ohlman. 2016. Attribute-Based Encryption on a Resource Constrained Sensor in an Information-Centric Network. In *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*. ACM, 217–218.

[12] N. Oualha and K. T. Nguyen. 2016. Lightweight Attribute-Based Encryption for the Internet of Things. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. 1–6. https://doi.org/10.1109/ICCCN.2016.7568538

[13] M. Özçelik, N. Chalabianloo, and G. Gür. 2017. Software-Defined Edge Defense Against IoT-Based DDoS. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*. 308–313. https://doi.org/10.1109/CIT.2017.61

[14] Amit Sahai and Brent Waters. 2005. Fuzzy Identity-Based Encryption. In *Advances in Cryptology – EUROCRYPT 2005*. Springer Berlin Heidelberg, Berlin, Heidelberg, 457–473.

[15] Reza Tourani, Travis Mick, Satyajayant Misra, and Gaurav Panwar. 2016. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Communications Surveys & Tutorials* 20 (2016).

[16] Xinlei Wang, Jianqing Zhang, and Eve M. Schooler. 2014. Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT. In *IEEE International Conference on Communications (ICC), 2014*. IEEE, 725–730.

[17] Xinlei Wang, Jianqing Zhang, Eve M Schooler, and Mihaela Ion. 2014. Performance evaluation of attribute-based encryption: Toward data privacy in the IoT. In *2014 IEEE International Conference on Communications (ICC)*. IEEE, 725–730.

[18] Meng Zhang, Hongbin Luo, and Hongke Zhang. 2015. A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials* 17, 3 (2015), 1473–1499.