Secure and Efficient Context Data Collection using Content-Centric Networking

Sin-seok Seo*, Joon-Myung Kang†, Alberto Leon-Garcia†, Yoonseon Han‡, and James Won-Ki Hong‡
*Dept. of Computer Science and Engineering, POSTECH, Korea
Email: sesise@postech.ac.kr
†Dept. of Electrical and Computer Engineering, Univ. of Toronto, Canada
Email: {joonmyung.kang, alberto.leongarcia}@utoronto.ca
‡Div. of IT Convergence Engineering, POSTECH, Korea
Email: {seon054, jwkhong}@postech.ac.kr

***Abstract*—Context data collection is a fundamental and important process for realizing context-aware recommender or personalization systems. The existing context data collection approaches are based-on traditional TCP/IP that has several disadvantages such as lack of mobility and security. On the other hand, Content-Centric Networking (CCN) provides advantages in terms of mobility, security, and bandwidth efficiency compared to TCP/IP. In this paper, we propose a secure and efficient context data collection and provision approach based on CCN. Simulation results show that this approach can reduce bandwidth consumption by 52.7%–98.9% in comparison to a TCP/IP-based one.**

***Index Terms*—CCN, Context Data Collection, Resource Management, Security**

I. Introduction

Various new types of context data are becoming available thanks to the improvements in sensing- and mobile-related technologies. Personalization and recommender systems can incorporate rich context-awareness by utilizing these plentiful context data. Accordingly, researchers have proposed architectures or methods to efficiently manage context data. A fundamental and important step for context management is context data collection. Traditionally, most context data collection approaches have used TCP/IP without consideration for mobility, security, or data transfer efficiency.

Content-Centric Networking (CCN) [1], a new approach for data communications that focuses on the content itself rather than the destination host, is a promising candidate for context data collection and provision because of the following advantages. First, CCN can reduce bandwidth consumption by providing long-term cache placed in a CCN node like Content Delivery Networks (CDNs). Second, CCN provides an inherent and flexible security mechanism that can be used to protect private and important context data. Third, CCN supports better mobility than traditional TCP/IP-based approaches; this is desirable for highly mobile context sources like Smartphones. Finally, CCN can reduce transfer delay because it does not require session establishment before actual data delivery occurs.

This research was partly supported by World Class University program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10100) and by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H0301-12-3002).

In this paper, we propose a secure and efficient method for context data collection and provision by applying CCN to our previously proposed context management architecture, U-CoUDE [2]. The new approach provides flexible and inherent security by encrypting every context data with three different types of symmetric key. It also provides efficiency in term of bandwidth consumption by taking advantage of CCN's content centric data transfer approach; simulation results show that our approach can reduce bandwidth consumption by 52.7%–98.9% compared to a TCP/IP-based one.

II. Background

This section introduces CCN's basic concepts and advantages in terms of bandwidth efficiency and data security. CCN [1] is an information or content centric data communication model proposed as an alternative to the traditional TCP/IP-based Internet. CCN is concerned about content itself for data delivery rather than the destination host that has the content. In CCN, there are two packet types: Interest and Content Object. The Interest packet is used for requesting content and it contains Content Name to identify the content. The Content Object packet is used for delivering the requested content and it contains Content Name, Signature, Signed Info, and the actual Data. A Content Object packet is transmitted only in response to an Interest packet, thus they are one-for-one and maintain a strict flow balance. To obtain content, a consumer broadcasts Interest packets with a Content Name. Any node that has the content that matches with the Name can respond with a Content Object packet.

Each CCN node maintains a Content Store, which acts as a long-term cache, to re-provide the contents that have previously passed through the node in response to requests that solicit the same contents. Accordingly, the more requests there exist for the same content, the more efficient CCN is in term of bandwidth consumption.

To provide protection and trust of content, CCN is built on the notion of *content-based security*. In CCN, data and their publisher's digital signature are encapsulated together

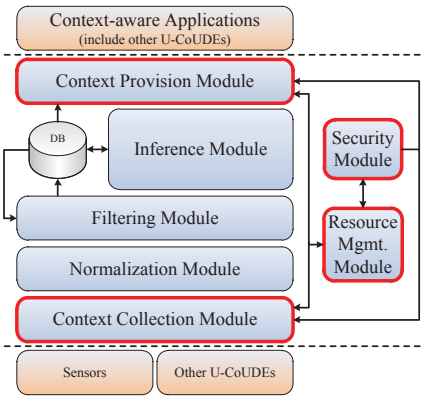

Fig. 1. U-CoUDE architecture.

within a Content Object packet to enable a data consumer to authenticate the publisher. In addition, private data are protected from unauthorized accesses by means of encryption. These features make possible CCN's dynamic content-caching capabilities. The protection level of the data can vary depending on the encryption method because CCN does not mandate a specific security or encryption model. The flexibility of the protection level allows CCN to accommodate diverse security requirements from various networking applications including context data collection.

III. Context Collection and Provision

In this section, we present our proposed context management architecture using CCN. We consider three approaches to security and discuss their advantages and disadvantages. In section IV, we compare their performance.

A. U-CoUDE

U-CoUDE [2], which stands for User-centric Context manager for Ubiquitous and Distributed Environments, provides basic concepts, a technology-neutral information model, and a high-level architecture to comprehensively manage distributed context data in a user-centric manner. Each U-CoUDE entity, as shown in Fig. 1, 1) collects context data from various sensors and other U-CoUDE entities (Context Collection), 2) translates different forms of context data into a standardized common form (Normalization), 3) filters and saves newly obtained context data by applying predefined rules (Filtering), 4) aggregates and associates context data to infer abstract and high-level contexts (Inference), and, finally, 5) provides collected and inferred context data to various types of context-aware applications including other U-CoUDE entities (Context Provision). A Resource Management Module maintains a list of available context sensors and registered U-CoUDE entities, and it provides the list to Context Collection and Provision Modules. A Security Module controls Context Collection and Provision Modules to protect private and sensitive context data of a user. In the architecture, Security, Resource Management, Context Collection, and Context Provision Modules are directly related to context collection and provision processes (bold lined boxes in Fig. 1).

TABLE I
Notations

Notation	Description
U	U-CoUDE entity
U^R	U-CoUDE entity that has a relationship with U
C	U-CoUDE entity that has a role of a context collector
P	U-CoUDE entity that has a role of a context provider
ID^U	Identifier of U
N^U	Human-readable name of U

A U-CoUDE entity has a role of either a context provider or a context collector when it collects or provides context data. It is also possible that the entity has roles of both a context provider and a context collector at the same time. Considering these concepts, Table I defines basic notations related to a U-CoUDE entity for describing reminder of the paper.

B. Security Module

Each context that is collected and provided by a U-CoUDE entity requires different levels of protection. For example, accurate location data of a user require high-level protection because they are very sensitive and private, whereas temperature data from a public place require low-level protection. To satisfy the different levels of security demands, U-CoUDE provides three *security types*: 1) Provider-based, 2) Group-based, and 3) Collector-based.

In the Provider-based security, context data are encrypted with a provider's symmetric key (sk^P) that is inherent to the provider and shared between the provider and its registered collectors; a sk^P is shared from a provider to a collector in a resource registration process (explained in section III-C). Therefore, every collector that is registered to the provider can access the Provider-based security context data by decrypting them using sk^P. As a result, the Provider-based security yields good bandwidth and delay performances because it can utilize CCN's data caching advantages, whereas it provides plain data protection because its symmetric key, sk^P, is shared with all of its registered collectors.

In the Collector-based security, context data are encrypted with a symmetric key (sk^{CP}) that is unique between a provider and a collector; a sk^{CP} is generated by a collector and shared with a provider in a resource registration process (explained in section III-C). Therefore, only a corresponding collector can access the Collector-based security context data that is encrypted with sk^{CP}. As a result, the Collector-based security provides the best data protection among the three security types, whereas it yields normal bandwidth and delay performances because it cannot utilize CCN's data caching advantages.

In the Group-based security, context data are encrypted with a group's symmetric key (sk^G); U-CoUDE entities can form a group and share the sk^G among them[1]. Only the group members can access the Group-based security context data

[1]U-CoUDE takes advantage of existing approaches for a group formation and its key management, such as Virtual Private Community (VPC) suggested in [3], rather than reinventing the wheel.

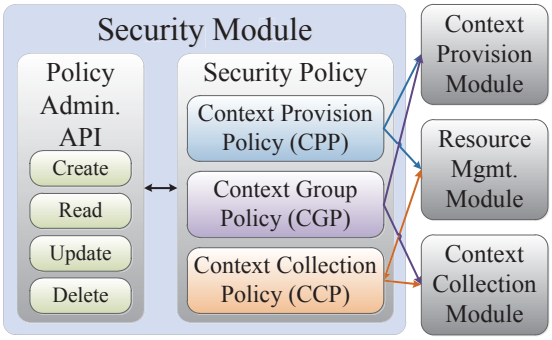

Fig. 2. Security module.

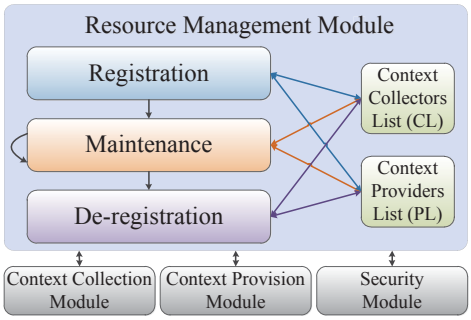

Fig. 3. Resource management module.

by decrypting them with the sk^G. This means that CCN's data caching will be effective only within the group members. As a result, the bandwidth and delay performances of the Group-based security are dependent on the number of group members, and it provides medium data protection compared to the other two security types.

The Security module shown in Fig. 2 contains two parts: Policy Administration API and Security Policy. The former provides the basic CRUD (Create, Read, Update, and Delete) operations to manage the Security Policy. The latter provides information for the three *security types* and consists of three sub-policies: Context Provision Policy (CPP), Context Collection Policy (CCP), and Context Group Policy (CGP).

The CPP controls the Resource Management Module to verify whether the resource-management-requests are coming from authorized collectors. It also controls the Context Provision Module to provide context data to only authorized and registered collectors. For the controls, the CPP contains the following information:

- **CPP-CL**: (ID^C, N^C, $domain^C$, pk^C), a list of trustworthy collectors, where pk^C is a public key of the C.
- **CPP-CT**: {*ContextType*, *SecurityType*, ($IDs^C_{auth}|IDs^G_{auth}$)}, a list of providable context types and their security meta-data, where IDs^C_{auth} and IDs^G_{auth} are lists of identifiers of collectors and groups that have the authority to collect the *ContextType* respectively; they are optional depending on the *SecurityType*.

The CCP controls the Resource Management Module to restrict the resource-management-requests to be generated and sent only to appropriate providers. It also controls the Context Collection Module to request context data only to appropriate and registered providers, and to verity whether pushed context data from providers are legitimate. For the controls, the CCP contains the following information:

- **CCP-PL**: (ID^P, N^P, $domain^P$, pk^P), a list of trustworthy providers, where pk^P is a public key of the P.
- **CCP-CT**: {ID^P, *ContextType*, *SecurityType*, (ID^G)}, a list of obtainable context types and their security meta-data per a provider, where ID^G is optional depending on the *SecurityType*[2].

The CGP provides group-related information to the Context Provision and Collection Modules. Based on the CGP, these two modules can decide whether a related U-CoUDE entity (U^R) is a legitimate member of the claiming group, or can decide which sk^G has to be used for en/decryption of context data in which case the *SecurityType* is Group-based. The CGP contains the following information:

- (ID^G, N^G, sk^G, IDs^G_{Mem}), a list of groups in which U is included, where ID^G is an identifier of a group G, N^G is a human-readable name of the G, sk^G is a symmetric key for en/decryption of context data, and IDs^G_{Mem} is a list of identifiers of U-CoUDE entities that are members of this group excluding ID^U itself.

C. Resource Management Module

Fig. 3 illustrates the three main processes of Resource Management Module and its interactions with the other modules. The main processes of the module include 1) Registration, 2) Maintenance, and 3) De-registration. Fig. 3 also shows both Context Collectors List (CL) and Context Providers List (PL) that are interacting with the three processes; they are lists of registered context collectors and providers to the U-CoUDE entity, respectively. More specifically, CL contains a list of (ID^C, sk^{CP}, α, β, γ) and PL contains a list of (ID^P, sk^{CP}, sk^P, α, β, γ), where α, β, and γ are availability-check-parameters (explained in section III-C2).

1) Resource registration: This is a process, from the perspective of a context collector C, for registering a context provider to collect context data from the provider. Similarly, from the perspective of a context provider P, it is a process for registering a context collector to provide context data to the collector. As a result of the registration process, C adds an item about the new P to PL, and P adds an item about the new C to CL. The detailed registration process follows.

1. C gets ID^P and pk^P from CCP-PL
2. C generates sk^{CP} and saves it to PL with the ID^P
3. C sends an Interest packet that has the following name to P

[2] ID^G has to be specified when the *SecurityType* is Group-based because U-CoUDE allows a collector, C, and a provider, P, to be grouped together for two or more groups at the same time. For example, C and P can form a group for a family while they are members of a co-workers group.

- */domain*/u-coude/ID^P/resourceManagement/registration/$E_{pk^P}(sk^{CP})$/$E_{sk^{CP}}(ID^C, DS^C)$/*nonce*

4. P gets sk^{CP} by decrypting $E_{pk^P}(sk^{CP})$
5. P gets ID^C and DS^C by decrypting $E_{sk^{CP}}(ID^C, DS^C)$ with the obtained sk^{CP}
6. P checks whether C is eligible to receive context data from P by referring to CPP-CL

A. If eligible,

- P saves the received sk^{CP} and ID^C to CL
- P sends a Content Object packet that contains $E_{sk^{CP}}(sk^P$, CPP-CT^{PC}, α, β, $\gamma)$ to C
- C decrypts $E_{sk^{CP}}(sk^P$, CPP-CT^{PC}, α, β, $\gamma)$ with the sk^{CP}
- C saves the sk^P, α, β, and γ to PL; and CPP-CT^{PC} to CCP-CT

B. If not,

- P sends a Content Object packet that contains a denial message to C
- C discards P-related information in PL

where DS^C is the C's digital signature and CPP-CT^{PC} is a part of P's CPP-CT that is directly related only to C. We adopt the notion of sending both sk^{CP}, which is encrypted with pk^P, and required information, which is encrypted with the sk^{CP}, as parts of an Interest packet's name from [4] (procedure 3–5). The *nonce* is added as the last part of the name to guarantee the Interest packet to be delivered to the intended P, *i.e.*, not to use CCN's data caching functionality.

After finishing the registration process, a collector maintains provider-related information in both PL and CCP-CT, and a provider maintains collector-related information in CL. Referring to the information, a collector collects context data from appropriate providers and a provider provides context data to eligible collectors.

2) Resource maintenance: This process deals with two aspects: *information update* and *availability check*. The information update is a process that a U-CoUDE entity notifies its information change to the registered U-CoUDE entities, and the registered entities modify their database with the notified information. The information update process and its relevant parameters vary depending on a process initiator. When the initiator is C, the relevant parameter is sk^{CP}, and when the initiator is P, the relevant parameters are sk^P, CPP-CT^{PC}, α, β, or γ.

The availability check is a process that a U-CoUDE entity checks the liveness of the registered U-CoUDE entities. It makes use of three parameters: α, β, and γ. The α is a time threshold for checking temporal disability of the registered U-CoUDE entity or the transport network. The β is a time threshold for determining de-registration. Finally, the γ is a time parameter used for periodically sending availability check message at the time in between α and β. They have to be determined by P and informed to C considering such as rates of context update, conditions of transport network, and importance of context data. The availability check process follows.

1. U maintains a timer τ for each registered U^R
2. The timer τ is set to 0 when a packet (either Interest or Content Object) is received from the U^R
3. If the τ exceeds α
   - U stops trying to collect/provide context data from/to U^R
   - U, then, periodically sends availability check message every γ to U^R
   - If the τ exceeds β, where $\beta > \alpha$, then U removes U^R from either CL or PL

3) Resource de-registration: This is a process, from the C's perspective, for removing an item of P in both PL and CCP-CT to stop collecting context data from the P. It is also a process, from the P's perspective, for removing an item of C in CL to stop providing context to the C. The de-registration process can be initiated by either C or P and the detailed process of each case is omitted in this paper because it is intuitive and obvious.

D. Context Collection and Provision Modules

In CCN, data transport is basically pull-based, *i.e.*, a data consumer requests data, then a data producer provides the requested data. A context data collection process in U-CoUDE corresponds with the basic data acquisition flow of CCN; C requests context data by sending an Interest packet containing the name that specifies the requesting context data to P and P replies to the Interest by sending a Content Object packet containing the requested context data. The detailed context collection and provision process follows.

1. C checks PL and CCP-CT to request appropriate context data; then, it sends an Interest packet containing the following name to P:
   - */domain*/u-coude/ID^P/contextProvision/pull/($ID^C|ID^G$/)*ContextType*/*TimeFrom*/*TimeTo*(/*nonce*)
2. P receives the Interest packet and checks CL and CPP-CT whether the context request is valid:

   A. If valid, P checks the availability of the requested context data

   A-a. If available,
   - P encrypts Context Value and its obtained Time (CVT) depending on the security type:
     - Provider-based: $E_{sk^P}(CVT)$
     - Collector-based: $E_{sk^{CP}}(CVT)$
     - Group-based: $E_{sk^G}(CVT)$
   - P sends a Content Object packet that contains the encrypted CVT to C
   - C receives the Content Object packet, decrypts the data, and saves the CVT

   A-b. If not available,
   - P sends a Content Object packet that contains an error message to C

   B. If not valid, P sends a Content Object packet that contains an error message to C

In the name of the Interest packet (of procedure 1), either ID^C or ID^G is required when the requested context data's

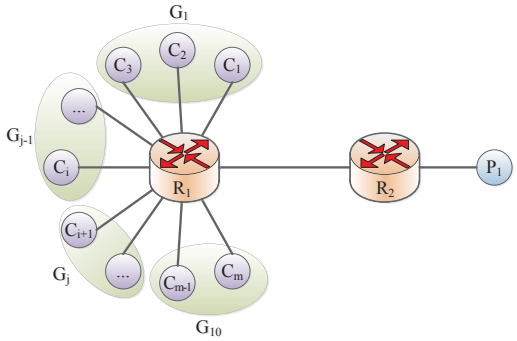

Fig. 4. Topology for the performance evaluation.

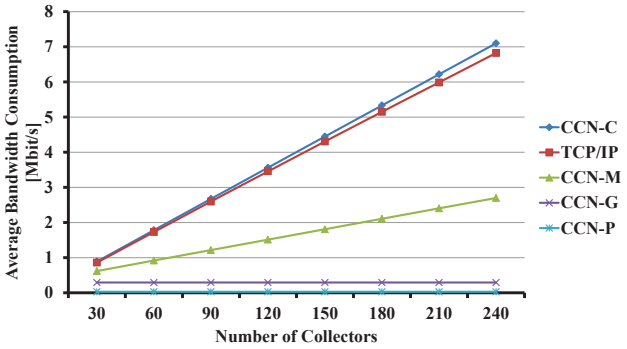

Fig. 5. Average bandwidth consumption (y) versus number of collectors (x) at the link between R_1 and R_2.

security type is Collector-based or Group-based, respectively. They are used by P to select the key to encrypt the context data. *TimeFrom* and *TimeTo* are time parameters for restricting the temporal range of the requesting context data. Finally, *nonce* is added to the name when the requesting Interest packet has to be delivered directly to the intended P without using CCN's data caching functionality.

IV. Performance Evaluation

We evaluated the performance of the proposed CCN-based context data collection approach in term of bandwidth consumption by comparing it to a traditional TCP/IP-based system. The evaluation was carried out in a simulation environment that was implemented using ndnSIM [5]. The ndnSIM is an open source module, which is designed to work with NS-3 simulator, for simulating CCN-based approaches.

In the simulation, we assumed the topology shown in Fig. 4; there are m collectors C_1–C_m, which are connected to the CCN router R_1, and one provider P_1, which is connected to the CCN router R_2. C_1–C_m and P_1 form ten groups (P_1 is not visually included as a group member for simplicity) and each group contains the same number of collectors as its members. We measured average bandwidth consumption at the link between R_1 and R_2 varying the number of collectors, in which case each C requested 5 context data per a second to P over 20 s and the average size of the returned context data from P was 512 bytes.

Fig. 5 shows the comparison of the measured average bandwidth consumption; CCN-P, CCN-G, and CCN-C represent cases where only one *security type* of Provider-, Group-, and Collector-based exists, respectively, and CCN-M represents a case where the proportion of the three *security types* was 1:1:1. CCN-P showed the lowest and steady average bandwidth consumption regardless of the increasing number of collectors. This is because the Provider-based security type can fully utilize CCN's caching advantages. CCN-G showed the second lowest and steady average bandwidth consumption similar to the CCN-P case. This performance metric is affected by the existing number of groups, which was fixed to ten during the simulation. CCN-M, TCP/IP, and CCN-C's average bandwidth consumptions were linearly increased following the growth of the number of collectors. Note that CCN-C consumed the most bandwidth because it cannot utilize CCN's caching advantages at all, like TCP/IP, and CCN's average packet overhead is slightly larger than TCP/IP's overhead. We insist that the slightly larger bandwidth consumption of CCN-C compared to TCP/IP is negligible considering additional advantages of CCN including security and mobility. Overall, our approach for collecting context data using CCN reduced average bandwidth consumption about 52.7% (CCN-M), 85.0% (CCN-G), and 98.9% (CCN-P) compared to TCP/IP.

V. Concluding Remarks

Context data collection is one of the most fundamental and important processes for realizing context-aware recommender or personalization systems. In this paper, we have proposed a context data collection and provision approach utilizing CCN. The proposed approach is secure because it encrypts every context data using a symmetric key that is shared between a context provider and a collector. In addition, the approach is efficient in term of bandwidth consumption because it utilizes CCN's caching advantages. In the simulation result, we have shown that our approach has dramatically reduced bandwidth consumption compared to TCP/IP-based one.

As future work, we will apply our approach to real-life applications including context change monitoring for managing Software Defined Networks (SDNs) and vehicle data collections for ITS.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT '09)*, Rome, Italy, Dec. 1–4, 2009, pp. 1–12.

[2] S. Seo, J.-M. Kang, Y. Han, and J. W.-K. Hong, "Context Management for User-centric Context-aware Services over Pervasive Networks," in *Proc. 14th Asia-Pacific Network Operations and Management Symposium (APNOMS '12)*, Seoul, Korea, Sep. 25-27, 2012, pp. 1–4.

[3] D. Kim and J. Lee, "CCN-based virtual private community for extended home media service," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 532–540, May 2011.

[4] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-over content-centric networks," in *Proc. ACM International Workshop on Re-Architecting the Internet (ReArch '09)*, Rome, Italy, Dec. 1, 2009, pp. 1–6.

[5] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," Named Data Networking (NDN) Project, Tech. Rep. NDN-0005, Rev. 2, Oct. 2012.