Towards Personal Content Networking

Joshua Joy
UCLA CS
jjoy@cs.ucla.edu

Young-Tae Noh
UCLA CS
ytnoh@cs.ucla.edu

Dae-Ki Cho
UCLA CS
dkcho@cs.ucla.edu

Uichin Lee
KAIST KSE
uclee@kaist.ac.kr

ABSTRACT

Recent advances of technology in consumer electronics have promoted a lifestyle where people live with convenience and ease by accessing any kind of information at their finger tips. These devices also allow people to generate/share a sheer amount of personal content such as photos, videos, and documents. However, personal content is now exploding, and personal content sharing/management is considered to be a challenging task particularly when users need to deal with personal content scattered over multiple devices. To mitigate this problem, we aim at enabling seamless access of personal content without specifying its location via content centric networking (CCN) over personal content. In this paper, we design a platform called personal content networking (PCN) that uses a single persistent, hierarchical naming space for personal content, allows users to securely initialize their devices and establish trust with other users, enables efficient content management over multiple devices (e.g., updates, removal, replication), and supports content centric access control via attribute-based encryption (ABE) for selective sharing where access control is not tied into hosts and yet fine-grained attribute based access control is permitted. We demonstrate its feasibility with prototype implementation on the basis of CCNx.

1. INTRODUCTION

Personal content is exploding. The storage demand appears to be infinite. A recent report estimated that by 2015, terabytes of data will be in a person's pocket, and petabytes of data in a person's home [1]. Under this circumstance, it is very important to have a system that seamlessly enables networking of personal content such that users can efficiently manage personal content scattered over multiple devices and selectively share content with friends.

Our work is motivated by the recent proposal of future Internet, namely content centric networking (CCN), a new Internet architecture that replaces conventional host-to-host conversations with named data oriented communications based on an URL-like persistent namespace [3]. The key features of CCN are that (1) it supports single persistent, hierarchical naming space and provides secure binding between name and data which can effectively thwart most security attacks; (2) it supports name based content access in which users can request content without specifying where the content is located, and any nodes that have the requested data can answer the request as nodes can cache data locally.

While building upon CCN, PCN addresses the unique issues that are essential to personal content networking scenarios. First, the system should provide an intuitive trust management mechanism for trustworthy content sharing among users (e.g., introducing other users into a personal network). Second, the system should support distributed content management over multiple devices such as replica management and content updates/consistency management. Third, the system must enable *content centric access control* for selective content sharing among friends. Simple public key based end-to-end encryption is not sufficient for access control, because this approach complicates content naming and nullifies the benefit of content caching in the intermediate nodes. Finally, the system should provide efficient content routing using an overlay network that is on the basis of social relationship.

2. PCN SYSTEM DESIGN

While building upon CCN, our system design aims to support the following features that are essential to realize personal content networking: (1) PCN uses a single hierarchical, persistent naming scheme of $N = P : L$ (where P is the name of a user, and L is the label representing the location of data in the hierarchy) and manages the trust using SPKI/SDSI [4]; (2) for secure identity introduction in mobile environments, PCN uses a secure introduction protocol that can effectively thwart the man-in-the-middle-attack; (3) given that one of the key functions of personal content networking is to share content among friends, PCN builds an overlay network based on social relationship; (4) PCN supports attribute based access control (ABAC) with attribute based encryption (ABE) to enable selective content sharing among users [5]; and (5) PCN users can effectively manage content in their personal devices and support content updates and automatic synchronization over CCN.

2.1 Naming

The current generation of personal devices use rigid and weak naming of the form “hostname:path.” The key problem is that content is tied to a host, making personal content management non-trivial, particularly when a user interacts with a number of devices (e.g., laptop, desktop, smartphone, ipad) and storage services (e.g., dropbox). A user has to track what files are in each of these devices/services and to decide how to migrate/replicate/update content.

Relationship among users in personal content networking is considered to be flat, and it is sufficient to use the public key as identity. Nonetheless there are cases where hierarchical naming is useful; e.g., a group of users has a set of subgroups. In SPKI/SDSI, a user can define a local namespace as a sequence of length two consisting of her key K followed by a single identifier (that is distinct within the local namespace). For instance, Alice with key K_a makes her own name as "K_a Alice." A study group with key K_g can name its sub-groups as "K_g sub1" and "K_g sub2." If a sub-group has multiple smaller groups inside, that group can name those groups similarly; e.g., sub1's two internal groups (ssg1 and ssg2) can be named as "K_g sub1 ssg1" and "K_g sub1 ssg2." Note that a local name is globally unique because the name contains a public key of the user. Moreover, each user can make signed statements of these local names, which allows anyone to certify a key via a web of trust (see the following section).

2.2 Device Initialization and Trust Management

As shown above, each PCN user has a private-public key pair which defines the user's name. When a new device is purchased, this information must be securely installed to initialize a PCN service. Moreover, for content sharing with others, a user must establish trust relationship by securely exchanging the public keys (e.g., how does Bob make sure that a key belongs to Alice?). For both problems (i.e., device initialization and trust establishment) secure key distribution is the main issue. Users can use USB sticks or can use local/wide area networks for key exchanges. The latter is less secure than the former, because it is vulnerable to the man-in-the-middle-attack—an attacker eavesdrops the channel and makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

2.3 Content Centric Access Control

We use attribute based encryption (ABE) with the goals of securely sharing content within a group across multiple untrusted servers and caches and yet of preventing collusion [5]. ABE is the key enabler for attribute based access control (ABAC) that supports fine-grained access policies. Each user first generates an ABE public key and an ABE master key. A user can define a set of attributes (e.g., college friends, CS219 team, family members) and an access policy using Boolean formula on attributes. This allows a user to perform fine grained access control: the user assigns attributes to each peer and then issues a secret key corresponding to the assigned attribute to the peer (using the master key). The user encrypts the file once based on the access policy, and any peer can decrypt the file if he has attributes that satisfy the policy.

2.4 Secure Replica Management

PCN allows users to replicate any files over multiple devices. Users can also browse the files scattered over multiple devices. Any files can be updated as long as a user has a right to update the content. Further, users can manage any files located in remote machines with "device-to-device communications" over CCN.

2.4.1 Synchronization

If a file is updated, a new version is created (with a new timestamp). To alert this event to the rest of nodes with the content, the node that makes the update will re-announce the corresponding prefix with a modification mark, which is a special type of prefix announcement for update notification. Each node will fetch the directory entries from the nodes that replicate the named prefix. From this, nodes can discover which files have updated and can perform file level synchronization.

Consistency: It is possible that a node may not be available at the time of an update announcement. In PCN, we support "eventual consistency" in which all replicas eventually converge to the same version given enough messages exchanged among participating devices (i.e., a file with the freshest timestamp) [7, 2, 6]. Eventual consistency is one of the widely used consistency models in disruption-prone mobile environments.

A node should be able to synchronize the files as long as it is connected to the overlay network. When a node re-joins the overlay network after disruption, it first checks its neighbors to find any missing prefix announcements. This allows the node to search for the updates of the files located in its local storage. If the node finds a prefix with a modification mark, it performs file synchronization as illustrated earlier.

PCN's two-phase process for synchronization is less efficient than the schemes in traditional distributed file systems. The notification via prefix re-announcement does not tell us where the update is originating from; it only hints that it is from one of multiple replicas. As a result, each node must probe all replica nodes (under the named prefix) to discover which file has updated, and then must fetch the updated file for synchronization. One simple solution that can mitigate this problem is to add an extra field in the prefix announcement that lists updated files and their version information (or a locator to the file that lists updated files with version information).

Conflict resolution: Due to disconnected operations, PCN should deal with conflicts where multiple devices modify the same file without knowing other devices' modification. If all the updates are properly applied in the order of modification time, we can simply perform an automatic merging process using the UNIX diff algorithm. However, automatic merging may fail under disconnected operations. In this case, PCN notifies the user that a conflict has detected. The user will be presented with a revision history including authors, dates, and versioned content. It is up to the user to resolve the conflicts and mark the content as merged.

2.4.2 Prefix protection

So far we assume that any node can replicate the content and announce the named prefix. After replicating the content, however, malicious users can launch an attack by inundating the network with fake update announcements. PCN nodes could waste considerable resources to handle such updates. Given that CCN does not deal with updates, this problem is unique to PCN.

To solve this problem, we propose to restrict that a prefix announcement is signed by the prefix owner. This mechanism is a reasonable approach in that people typically want to have a full control of their namespace and the locations of files in a multi-device environment (e.g., all document files in one's laptop). A similar technique is used in BGP security where each prefix is signed in order to prevent prefix hijacking where an attacker has a partial or full control of the named prefix. While this problem is less serious in CCN because requests (interests) will be routed to all replicas and a denial of services cannot be achieved, it is still possible that malicious users can launch update flooding attacks.

2.4.3 Update access control

PCN basically assumes that a namespace owner can manipulate the file as he wishes (e.g., as in Microsoft Windows). We can also implement a more sophisticated access control mechanism, e.g., UNIX like access control with read-only or read-write access rights. To this end, we maintain an ACL file in each directory that details access rights on files therein (say ".acl"), which is signed by the owner. The ACL file can be replicated as a regular file and can be securely distributed over the network. Replica nodes can then use this ACL file to decide whether to permit an update. The ACL file can be encrypted to protect an owner's privacy (e.g., using ABE). It is up to the user how to set up one's own personal content networking system.

2.4.4 Replica management

A user may want to know what files are stored where and wish to replicate files to remote devices. Regular content browsing like UNIX command *ls* does not tell users in which device the files are located. For replica management, PCN reserves a special file located in each replicated directory, namely ".replica" that contains the information about the replicated files in the named directory and the information about the device. In PCN, we reserve a special directory for devices, namely the "/dev" directory through which a user can freely name personal devices. For instance, Alice's iPad can be named as "/Alice/dev/iPad." Further, each device announces this device name prefix, which will enable device-to-device communications over CCN.

A user can list the files and their replication status over the remote devices by simply collecting ".replica" files. This procedure is very similar to content browsing except that we are retrieving all the ".replica" files within the named prefix. For instance, Alice can check the replication status of her music files by fetching "/Alice/my music/.replica." If the directory contains sub-directories, the user needs to recursively retrieve sub-directories to know replication status. Note that a user may not wish to reveal the replication status to other people as it may be considered personal. In this case, the user can encrypt ".replica" files using ABE and assign attributes like "personal items" to ensure that he can only know the replication status.

3. CONCLUSION

Our goal has been realizing personal content networking (PCN) that builds on top of existing content centric networking (CCN). In PCN, each user has a single persistent namespace of personal content that can significantly reduce the burden of content access/management across multiple devices/users. Users can securely initialize their devices and establish trust relationships with other users via a secure introduction mechanism. Resulting trust relationships are then used to construct an overlay network on which CCN is used for content delivery. Given that the decentralized nature of content delivery in CCN makes traditional host centric access control challenging, we proposed content centric access control where access control is not tied into hosts. We realized content centric access control via attribute-based encryption (ABE) where fine-grained access policies of attribute based access control (ABAC) are carried along with encrypted data for decentralized access control. Additionally, we have built a PCN prototype on the basis of CCN.

4. REFERENCES

[1] Personal Content and Home Network Storage, the Perfect Storm, Tom Coughlin, 2007.

[2] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. Coda: A Highly Available File System for a Distributed Workstation Environment. IEEE Transaction on Computer, 39(4):447âĂŞ459, 1990.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking Named Content. In

[4] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate Chain Discovery in SPKI/SDSI. Journal of Computer Security, 9(4):285âĂŞ322, 2001.

[5] J.Bethencourt, A. Sahai, and B. Waters. Ciphertext-Policy Attribute-Based Encryption. In SP'07, Oakland, CA, Apr. 2007.

[6] P. Reiher, J. S. Heidemann, D. Ratner, G. Skinner, and G. J. Popek. Resolving File Conflicts in the Ficus File System. In USENIXâĂŹ94, Boston, MA, June 2004.

[7] J. M. Paluska, D. Saff, T. Yeh, and K. Chen. Footloose: A Case for Physical Eventual Consistency and Selective Conflict Resolution. In WMCSAâĂŹ04, Lake District National Park, UK, Dec. 2004.