ICN Research Group X. Zhang
Internet-Draft R. Ravindran
Intended status: Informational Huawei Technologies
Expires: August 25, 2013 H. Xie
Huawei & USTC
G. Wang
Huawei Technologies
February 21, 2013

PID: A Generic Naming Schema for Information-centric Network
draft-zhang-icnrg-pid-naming-scheme-02

Abstract

In Information-centric network (ICN), everything is an identifiable object with a name such as a named data chunk. Different from host-centric connectivity, ICN connects named entities using name-based routing and forwarding. At the same time, network entities, end devices, and applications have variant demands to verify the integrity and authenticity of these entities through names. This document proposes a generic naming schema, called PID, which supports trust provenance, content lookup, routing, and inter-domain resolution for ICN. With PID schema, a name consists of three components: principal(s), identifier(s), and domain(s). In this draft, we only illustrate the principles and concepts of PID and the functional role of each component, and leave encoding approaches as implementation options.

Status of this Memo

Copyright Notice

Table of Contents

1. Design Principles

1.1. Naming in ICN

In ICN design, a name has been required to serve for many purposes: ICN requires unique names to identify mutable or immutable content or information objects; in data caching, a name is used to look up and access the data; in routing and forwarding, a name is used for reaching the information object; for security, a provenance between name and data is established and verified via cryptographic credentials associated with a name. We summarize the following roles that a name may be desired from different users or stakeholder in ICN:

- R1 (unique): A name identifies an object or entity with uniqueness in some scope (e.g., within a domain or Internet).

- R2 (locatable): A name enables interested entities to locate the identified object in a network. For this purpose, the name is either routable to reach the object, or includes information to derive the routable location(s) of the object.

- R3 (readable): A name enables a user or application to easily identify and indicate the content of an object, even without knowing the content itself beforehand or before the content is generated. For this, the name may be required to be human-readable.

- R4 (authenticable): A name has strong binding with the content itself (either the publisher or owner of the content, or the content itself), in order to provide content access authentication, to let receiver verify the provenance, and to prevent denial-of-service attacks in an ICN [ICN-name].

- R5 (trustable): A name includes information on how to derive the trust of a content object, e.g., by an end user who retrieves the content from ICN. The trust can be built on mechanisms out of ICN primitives.

There may be many different naming schemes towards all or subset of the above roles. For example, flat names are used in DONA [1] and NetInf [2] for global uniqueness and authentication, but does not provide readability, routing, and trust-deriving information. PSIRP and PURSUIT [3] sperates the namespaces for rendezvous and forwarding identifiers of a name, and both namespaces are flat. Standard ways to name objects with hash functions have been proposed in [4], where a named identifier (ni) scheme is used to uniquely specify object. This name schema focuses on the uniqueness and strong binding of content and name, but not for routing. Hierarchical flat name is proposed in [5] to use nested flat names for routing purposes. Hierarchical human-readable names are proposed in CCN and NDN [6], but they do not provide authentication and trust-deriving information. A generalized form of name is proposed in [7] to bind authentication with content names via a signature.

1.2. Design Principles for Naming in ICN

We follow several principles for defining naming schema in ICN.

- A naming schema satisfies necessary but not more than necessary aforementioned roles: in our view, a single-component name cannot satisfy all roles at the same time.

- A content name identifies a content object in persistent way, such that this name does not change with the mobility and multi-home of corresponding content, device, or host. A client can always use this name to retrieve the content from network and verify the binding of the content and the name.

- A naming schema should give certain level of flexibility to support different networks, considering variant network architectures have been proposed and multiple ICNs and current Internet may co-exist. Ideally, a name can include any form of identifier, including flat, hierarchical, human readable or non-readable. The identifier can be chosen by content owner or publisher with the uniqueness within certain domain or within an application-specific scope.

- The network does not use persistent content name for routing directly; instead, a "routing name" (or routable address/location/label/tag) is network architecture dependent, which is usually routable within the network, such that a network node or client can reach the content with it. Usually, a routing name is the real location (or locator) of the content in the network.

- Per-domain-based (globally or locally) naming resolution services (NRS) should be available, to map a persistent content name to routing name or location. While per-domain NRS updates the routing labels for a content name, it creates a late-binding routing behavior. We note that a single content name can be mapped to multiple routing names. How to implement name resolution service is not included in this draft, e.g., [8] provides details of one implementation.

2. PID Naming Schema

2.1. Naming Format

Based on these principles, we propose a P:I:D (or simply PID) naming schema for ICN. Each name is specified by three components of PID, where:

- P is the principal to bind the object with complete name for security purpose, for different relationships, e.g., ownership, administration, and social relations. P is usually constructed by hashing the public key of the principal, or hashing the content object itself if it is static. We call the relationship between P and the object as "security binding".

- I is the identifier of the object in variant forms and is referred by end user, applications, or other entities. It can be something chosen by publisher or a network service, or other administrative authorities. It can be hierarchical or flat, user-readable or non-readable, and usually location-independent. We call the relationship between I and the object as "application binding".

- D is the domain that provides resolution from identifier to the real location of the object by routers. For persistence purpose, D can be in any of the following forms:

  * The locator of the target object if the locator is persistent;

  * A resolution service name or location which maps the content identifier (I) to its real location, if the resolution service name is persistent;

  * A resolution service name that maps the content identifier (I) to another resolution service name or location, that is, a meta-domain;

  * Any combinations of above.

  We call the relationship between D and the content object as "network binding".

For example, D can be the domain name of the publisher’s domain gateway, service, host that can resolve P:I, or a redirection gateway, service or host to preserve name persistence or to deal with mobility or hosted services. D is the "fall back" used for name-resolution if P:I is not resolvable in the local cache of the requesting domain. D is usually routable (globally or locally), such that, when an application or network node first receives an interest with the content name, it can query a resolution service by routing with D and obtain the real location or locator of the named object. In case the resolution service is not static, a recursive name resolution may be performed, i.e., D points to a static resolution service, which in turn points to a dynamic resolution service, which points to the location of the object. If there is no D in a name, then a network node uses I to route to the location of the object if I is routable.

D can be in the same namespace of I, but in general it can be different. For example, in one case, D is the container of a set of objects which can locate and resolve objects [9].

For a published name that is in PID scheme, a change of any field in P or I or D re-names the object, e.g., the object is re-signed by another entity, or its resolution service is changed, e.g., the publisher changes the host service of a web page.

We note that the domain concept in our naming schema is more general than the administration domain in current Internet architecture. In PID, the relationship between a named object and its domain D is for location resolution and routing purpose. It can be the same as the administration domain of the content object, or a 3rd party resolution service provider, where the designated domain provides resolution service. In more general way, the domain of a name can have social-, admin-, owner-, host- relationships with the named object, which implies that the domain provides resolution service to locate a content object with its name. A domain can provide a DNS-like service that maps a content identifier to the location of the object or the resolution service. Different from current Internet's centralized DNS, a domain-based resolution can be more general with a distributed implementation. Furthermore, the meta-domain of a content object can be personal profile, e.g., as in social network service, an enterprise directory service, a cloud service provider, or a web hosting service. For example, to support the Example 2 of [10], the domain part of the content name is simply the service name or location of the lookup database, which is more persistent than the mapping of a content identifier to location. Note that in [10], the lookup database is assumed to be static and already known by the network, which we believe is not realistic and flexible enough.

2.2. Routing Names

As aforementioned, our naming schema differentiates content names and routing names, where the former is persistent to specify a content object, while the later is location-based for routing purpose. Instead of a very specific format of routing names, our schema supports variant routable names (or routing labels), e.g., a network address or a locator. For a content name P:I:D, the D resolves P:I to one or many routing labels, and application or network router can choose one to reach the content or more for multicast. A routing label for a content object can be dynamic, and can be changed from domain to domain. For example, a single domain may by default set a gateway routing label to all the clients it is serving. The gateway may then replace it with some other label. Through this way, the routing label can allow policy-based intra/inter-domain routing, late binding for mobility, and delay-tolerant content routing.

With a content name provided by a content requester, the network first returns the real location of the named object via resolution services specified by the domain information (D) in the name. This location information is then augmented in the head field of a PDU (e.g., an interest in CCN). The network then uses this location information to reach the object, retrieve the named content, and forward back to the requester. Resolving the location from name and augmenting the PDU can be transparent to applications.

In general with P:I:D the resolution process works as follows: with a content name P:I:D, a client forwards request to a network node (e.g., an access router), if not resolvable in the local cache, the router first routes to a naming resolution service (NRS) with D. With the input of P:I, the NRS returns the routing name (or routing label) of the content object, e.g., a location or a locator. We note that the format and semantics of a routing name can be domain specific, and may be only routable in one domain, e.g., it can be a flat location in DHT or a hierarchical node name in a network operator. Upon receiving this, the network node inserts this label in the head of the interest packet. The network then uses this routing label to reach the next hop, to retrieve the named content by using P:I at each hop, and to forward data back to the requester, e.g., following the PITs in CCN [6]. In case the routing name resolved from the NRS is another name resolution service named with D’, the network node sends the request to this revolved NRS with D’ in interest head, obtains the location of the target object, and then inserts the location into interest head to obtain the content object. This process happens recursively until the location of the named object can be reached. With a single name, an NRS may return multiple entries of the locations of object. A network node can use one or multiple of them to retrieve the object, according to its local policy or configuration. In another case, where a separate locator address space is not managed, a per-hop forwarding can be adopted, where a content router tries to resolve the content name identifier (I or P:I) locally in its cache, if it is un-resolvable, use I:D or just D to route to domain D, in the latter case once the interest reaches D, the request I:D can be used to route to location(s) of the content object.

Therefore, logically, a data PDU could be of form <P:I:D, <Routing Label>, C, Sign_P(I:D,C), Metadata >, where C is the content payload, Sign_P is a signature generated from the private key corresponding to P on C and persistent content name, and the metadata includes other meta attribute information. With this hybrid naming approach, our schema achieves the benefits of both pure self-certified names and hierarchical names. Specifically, similar to hierarchical human-readable name, the P:I part of our name schema can achieve global uniqueness and readability (if needed). With D, our name schema achieves location persistence without including the real location of the content in name. With the P part, our name schema can achieve strong binding between content and its name for security and data integrity. Note that trust management is usually built on some external mechanism out of the naming schema.

In a special case, the D of a content name P:I:D could also serve as a routing label, i.e., D can serve dual purposes: a resolution/ redirection point, and a routing label as well, e.g., D could directly resolve to a container (server). This avoids one RTT to obtain the Routing Labels of the content name.

While D can serve the same purpose of routing label that is proposed in [10], our PID schema has two improvements:

- P:I:D has better persistence property since it separates routing labels from content names, while in [10], a content ID includes both routing labels and identifier. When the routing label of a content object is changed, e.g., the host service is changed, or a new host service is added, the content ID has to be changed, which destroys the name persistency.

- P:I:D has stronger security binding of name and content via principal field.

Note: We focus on the logical semantics of fields in a naming in this document. In implementation, variant formats of P:I:D can be options. For example, I:D can be in a single component, which acts as a resolvable identifier.

2.3. Cache Access

With a content name of P:I:D, a router can use the full name to index and look up cached content chunks and pending interests, e.g., in content sore (CS) and pending interest table (PIT) in [6]. Optionally, a router can only use P and I for the same purposes. This achieves location independence in data storage and forwarding, e.g., when a content chunk with P and I can satisfy any request of P:I:D with any D. That is, two content objects with same P and I are considered as the same and thus only one is cached at anytime, even though they may have different Ds.

2.4. Dynamic Content Routing

The P:I:D naming lends itself to allow consuming and producing applications to choose naming semantic that meets requirements in terms of reliability, security or performance metrics. The naming format follows a P:I:D format, where I identifies the named entity with a local or global scope, and D is the authority which could resolve the entity’s location(s), and P securely binds the content object to I. For content routing I:D is the relevant portion. As I could be a hierarchical or flat name, several options for content routing are possible. In one case separate ICN domains can be built that are optimized to deal with either flat or hierarchical, where name-resolution service allows the request to be directed to the appropriate domain criterion determined by the publisher, consumer or based on certain routing policies. In another case, a content routing domain can be built where the name-resolution infrastructure is enabled to deal with both flat and hierarchical names, where irrespective of the type of naming, a separate locator space exists to resolve the content name to its location(s).

If the combination of I:D is hierarchical, the content routing can follow the resolution mechanism similar to CCN. To resolve an interest, either I itself could be routable if it is globally unique, or the combination of I:D should be routable, which shall be interpretable by the name resolution service handling hierarchical names. Such ICN domains can leverage longest prefix match to take advantage of name-prefix aggregation mitigating routing scalability issue.

If I is flat, then the resolution through D should return a routing label(s), which can be appended to the interest packet for intra- and inter-domain name based routing on a fast path, or the name resolution can be handled by the global name resolution infrastructure through inter-domain cooperation on a slow path.

There are several considerations for dynamic name based routing. Based on the particular naming construct, hierarchical vs flat vs hybrid each of these considerations achieves the same objectives respectively with different mechanisms.

2.5. Towards Generic Naming Schema

As mentioned before, one object may have several names. Different names are assigned from different domains and served for different purposes. Logically, for a single object (e.g., a content, a device, an application, a service, a network nodes, or a user), it can have multiple identifiers, For example, a mobile device may have identifier of IMEI, a phone number, an IP address, a human readable name (e.g., Alice's iPhone), and an organizational device id (e.g., if the device belongs to a company). A user generated content can have a user chosen ID, a URL, and a tinyURL. All these identifiers can have a single principal. Therefore the name of the object can be P:(I1:...:In):D, where Ix is an identifier, D is a domain that provides name resolution service, and P is the principal.

In very general case, each identifier can be associated with different principals, and multiple locators can be used for a single content object, e.g., for load balance and duplication. For example, the Abel's iPhone have different public keys for different names it may use for different network services, one for Abel's personal use, and another from the enterprise. Therefore, the relationships between the object, identifier, and principals can be illustrated as follows.

As one object may have many persistent domains (e.g., a content is stored at different host services or CDNs), and one object may also have many IDs, in this generic schema, both domain and identifier may be a multi-element set, and content routers and consumers can select variant elements for content routing and forwarding (based on locally defined policy).

Note that there can be mapping relationships between multiple names of a single object. For example, an object may have a hierarchical identifier within its local domain owned by an enterprise, but has a flat identifier (hash of its content) with a DHT service. There can be a mapping service to link these two names towards the same object.

In general, mapping function between different names of a single object can be used to build flexible relationships between names, such as:

o An identifier can be derived from another identifier, which forms nested or tunneled names.

o A principal can be signed by another principal, to build trust between different principals, such as for ownership, administration, and social relationships.

o A domain name can point to another domain name for the same object.

The P:I:D schema can support these levels of flexibility. However, we consider these are extensions of core naming schema.

3. Security and Trust Management

As traditional, the integrity of a content object is maintained by the signature included in each data chunk. If the principal (P) of the object name is the public key or hash of the public key of its publisher, this key can be used to verify the integrity and authenticity of the object. When P is the hash of the content itself, then the signature itself is built with P. Therefore, PID provides a strong binding between the name and the content of the object.

When P is (the hash of) a public key, it can be the trust derivation information of the object, e.g., an end user can use it to decide whether to trust the content or not, based on some trust management infrastructure such as PKI or name-based trust [11]. However, PID schema is independent from any trust management infrastructure. The trust of a content object is derived from the trust of the principal. Either network nodes or end users can verify the trust of a content object. The trust management infrastructure is out of the scope of PID naming schema.

Similar to [6], the public key of a principal can be regular ICN data, also with the name of P:I:D. For the name of a certified public key, its I can be some domain- or realm-based name, D can be the name (if static) of the certificate directory service of a CA, or a domain that resolves the location of a public key certificate, and the P is the hash the CA’s public key.

4. Security Considerations

No further security issues are not discussed in this memo.

5. IANA Considerations

This document makes no specific request of IANA.

6. Conclusions

In this draft, we propose PID, a naming schema for ICN. With this schema, an object name includes a principal P, an identifier I, and a domain D. The principal P acts for security binding, e.g., to verify if the object is bounded with its name, and to derive the trust of the object with possible trust management mechanisms. The identifier I identifies the object within certain scope, and can be used for application binding such as caching access. The D refers to a name resolution service that can resolve the real time location of the object, directly or recursively. While this draft lays out the basic design principles and workflows of PID, we leave its encoding and implementation options to other documentations, such as [9].

7. Informative References

[1] I. Koponen et al., "A Data-Oriented (and Beyond) Network Architecture.", Proc. of ACM SIGCOMM 2007.

[2] C. Dannewitz et al., "Secure Naming for a Network of Information.", IEEE INFOCOM Computer Communications Workshops 2010.

[3] PURSUIT, "http://www.fp7-pursuit.eu".

[4] S. Farrell et al., "Naming Things with Hashes.", http://datatracker.ietf.org/doc/draft-farrell-decade-ni/ 2012.

[5] A. Ghodsi et al., "Naming in Content-Oriented Architectures.", Proc. of ACM ICN Workshop 2011.

[6] V. Jacobson et al., "Networking named content.", Proc. of ACM CoNEXT 2009.

[7] D. Smetters and V. Jacobson, "Securing Network Content.", PARC Technical Report 2009.

[8] R. Wang et al., "Container Resolution System in ICN.", http://datatracker.ietf.org/doc/draft-wang-icnrg-container-resolution-system/ 2013.

[9] C. Yao et al., "Container Assisted Naming and Routing for ICN.", http://www.ietf.org/internet-drafts/draft-yao-icnrg-naming-routing-00.txt. 2013.

[10] A. Narayanan and D. Oran, "NDN and IP Routing, Can it Scale?", http://trac.tools.ietf.org/group/irtf/trac/raw-attachment/wiki/icnrg/IRTF%20-%20CCN%20And%20IP%20Routing%20-%202.pdf 2011.

[11] X. Zhang et al., "Towards name-based trust and security for content-centric network.", Proc. of IEEE ICNP 2011.

Authors' Addresses

Xinwen Zhang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone:
Email: xinwen.zhang@huawei.com

Ravi Ravindran
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone:
Email: ravi.ravindran@huawei.com

Haiyong Xie
Huawei & USTC
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone:
Email: haiyong.xie@huawei.com

Guoqiang Wang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone:
Email: gq.wang@huawei.com