Towards Name-based Trust and Security for Content-centric Network

Xinwen Zhang[1], Katharine Chang[2], Huijun Xiong[3], Yonggang Wen[4], Guangyu Shi[1], Guoqiang Wang[1]
[1] Huawei Research Center, Santa Clara, CA, USA, {xinwen.zhang, shiguangyu, gq.wang}@huawei.com
[2] University of Michigan, Ann Arbor, MI, USA, katchang@eecs.umich.edu
[3] Virginia Tech, Blacksburg, VA, USA huijun@cs.vt.edu
[4] Nanyang Technological University, Singapore, ygwen@ntu.edu.sg

***Abstract*—Trust and security have been considered as built-in properties for future Internet architecture. Leveraging the concept of named content in recently proposed information centric network, we propose a *name-based* trust and security protection mechanism. Our scheme is built with identity-based cryptography (IBC), where the identity of a user or device can act as a public key string. Uniquely, in named content network such as content-centric network (CCN), a content name or its prefixes can be used as public identities, with which content integrity and authenticity can be achieved with IBC algorithms. The trust of a content is seamlessly integrated with the verification of the content's integrity and authenticity with its name or prefix, instead of the public key certificate of its publisher. In addition, flexible confidentiality protection is enabled between content publishers and consumers. For scalable deployment purpose, we further propose to use a *hybrid* scheme combined with traditional public-key infrastructure (PKI) and IBC. We have implemented this scheme with CCNx open source project on Android.**

I. Introduction

Information centric networks (ICN) such as content-centric network (CCN) [9]and named data network (NDN) [18] treat named content as the first class citizen: content publishing, requesting, managing (modification, deletion, etc.), and reachability are all determined by content name, rather than digital address of host machines in traditional IP networks. However, both name-centric principal and access flexibility arise new challenges in fulfilling the built-in security requirements, which are essential for next generation Internet infrastructure. An intrinsic demanding is how to bootstrap trust and enable secure communication between content publishers and users in content network.

CCN and NDN have proposed basic mechanisms for content integrity and authentication verification. For example, content should be digitally signed by its publisher's private key such that the integrity of the content can be verified later using the publisher's public key by its consumers or network routers at the time they receive the data from the publisher or other routers. A related problem is content confidentiality which requires only authorized users can access the content. Usually, this can be achieved via symmetric encryption on data and secure distribution of symmetric key to only legitimate users. A typical approach for secure key distribution is based on public/private key pairs of users, in which the encrypted session (symmetric) key for content is encrypted by a receiver's public key, which in turn can only be decrypted by its private key. Overall, a certificate authority is needed to assure the ground trust of the public/private key infrastructure, which is not practical in reality. Especially in the content network scenario, user level key and certificate management introduces high cost for many large organizations and inconvenience for sharing the same keys among multiple devices of a single user.

We propose a *name-based* mechanism for efficient and flexible trust management and secure communication. Our proposal is built on top of identity-based cryptography (IBC), which is a type of public-key cryptography. In IBC, a public identity (e.g., a string of email address, phone number, or a hierarchical identity within an organization) acts as a public key. Therefore a digital signature can be verified by the identity, and any data encrypted by the identity can only be decrypted by the private key correspond to this public identity. Thus, IBC eliminates the certificate management in public key infrastructure (PKI). Furthermore, to send secret data to a remote party (e.g., data encryption key), the sender does not need to obtain and verify the public key certificate of the receiver before sending; instead, it can use the public identity of the receiver to encrypt the data such that if the receiver wants to read the data, it has to obtain a private key with its identity from the PKG.

These features make IBC a flexible solution for name-based trust and security in ICN. First, a content name or prefix can act as a public identity, such as `nytimes.com/`$today$`-paper`, where $today$ can be a real date. and each receiver can verify the integrity and authenticity of the content by the using the content name directly without obtaining the public key certificate of the host server and possibly multiple CA's certificates. Through this, the trust of the content is directly derived from the human-readable content name or prefix instead of indirectly from certificate chains. Second, for the sake of confidentiality, a publisher can publish sensitive contents encrypted with identities which either it trusts (e.g., a destination's email address or phone number) or a content name selected by itself (e.g., via a name registration service). A receiver can decrypt the content only if it has the corresponding private key.

IBC algorithm requires a trusted private key generator (PKG) which generates public system parameters for data encryption and signature verification, and extracts private keys for users. With this centralized party, simply applying IBC in

Internet-scale content network will have the issues of scalability and incremental deployment, especially trusting a single or few PKGs for all Internet users. Towards these challenges and motivated from previous work [15], we propose to use a *hybrid* scheme by combining PKI and IBC. Specifically, a PKG generates and distributes private keys for clients in its local domain or autonomous system (AS), while existing deployed PKI-based trust management infrastructure (e.g., that for DNSSEC or IPSec) can be leveraged for certifying domain level PKG's public system parameters such that an end content consumer only needs to trust its PKGs and very few CAs. With this, we claim that PKI-based scheme is good for trust management between domains, while IBC-based scheme fits better for trust management for end users or devices as trust is derived from known identities. Our proposed scheme leverages the advantages of both, which we believe fits the convergence of security and trust requirements in both application level and network level for content network. In some cases where a domain-based PKG is not viable, our scheme supports self PKG: an individual end user can be the PKG to generate private key and parameters, which has been demonstrated to be viable in large scale applications such as online social networks [3].

This paper first introduces the IBC algorithm briefly, and then presents our solution with IBC and PKI in content network with some deployment discussions. We then depict our prototype implementation with CCNx on Android device for secure personal data sharing over content network, and highlight some related work. Finally we summarize our ongoing work towards general security and privacy considerations.

II. Cryptographic Primitive

In an IBC system, a private key generator (PKG) generates a master secret key (MSK) and public system parameters (SP). The MSK is kept as a secret and used by the PKG only to generate corresponding private keys for individual users, and SP is published. Any user can use the published SP and the publicly known user identity to generate public keys for other users. In what follows, we recap some basic algorithms of IBS [14] and IBE [5], which we will apply in our scheme.

Identity-based Signature An identity-based signature (IBS) scheme consists of four algorithms: `Setup`, `Extract`, `Sign`, and `Verify`. There are three parties in the system, the PKG, a signer, and a verifier.

- `Setup`(1^k): The PKG runs this algorithm that takes in a security parameter 1^k as input and generates the master secret key (MSK) and public system parameters (SP). The MSK is kept secret by the PKG and SP is made public.
- `Extract`(MSK, ID): The PKG runs this algorithm that takes in the MSK and an user identity ID as inputs. The algorithm generates a secret (private) key (SK) for the corresponding user ID.
- `Sign`(SK, M): A signer runs this algorithm that takes the SK of the signer and a message M as inputs and generates a signature σ of M.
- `Verify`(SP, ID, M, σ): A verifier runs this algorithm that takes in the SP, the signer's identity ID, the message M, and the corresponding signature σ received from the signer as inputs and returns 1 if σ is valid for ID and M, and returns 0 otherwise.

Identity-Based Encryption An identity-based encryption (IBE) scheme consists of four algorithms: `Setup`, `Extract`, `Encrypt`, and `Decrypt`. There are also three parties in the system, the PKG, an encrypter, and a decrypter. The `Setup` and `Extract` algorithms are the same as in IBS performed by the PKG.

- `Encrypt`(SP, ID, M): An encrypter runs this algorithm that takes in the SP, the receiver's identity ID, and a message M as inputs and generates a ciphertext C of M.
- `Decrypt`(SK, C): A decrypter runs this algorithm that takes in its SK and the ciphertext C received from the encrypter as inputs and returns the message M.

III. Our Scheme

A. Name-based Trust with IBS

As aforementioned, one challenging problem in ICN is to derive trust of a received content for a consumer. Without evaluating the content itself, trust is mainly obtained from the credentials of its publisher, in particular, its public key certificate. With traditional PKI-based trust management scheme, the trust is indirectly associated with that to a CA or multiple CAs, which may or may not have direct trust relationship with a user. One design principle that we take is to derive trust directly from publicly known identities or content names, by following the overall architectural principle of "security must be built into the architecture" [18]. We note that in the real world, trust between human beings are built with known identifies of each other, which are usually provided by physical or social infrastructures and relationships (e.g., phone numbers, family members, social groups, etc) or even Internet services (e.g., email address), which people trust them in common sense. We believe that bringing these trust relationships to content network can enhance the trustworthiness of published content. That is, we need some link between contents and identities, such that if we can trust the identity, we have some level of trust for the associated content. Furthermore, we wish this link be easily verified, i.e., along with the integrity verification of the content by a consumer.

Two approaches can be taken with IBS for these purposes. First, the identity is that of a content's owner or provider. When the content is signed by its provider with IBS, a consumer can verify the signature with the identity of the provider, and if successes, the consumer has the assurance that (1) the content's integrity is preserved – it is not modified after being published by the provider; (2) the content is signed by someone who owns the identity. With this, the trust of the content is derived from that of the content provider to the consumer, e.g., the data is published by someone in her phone contact list. For the second approach, the identity can

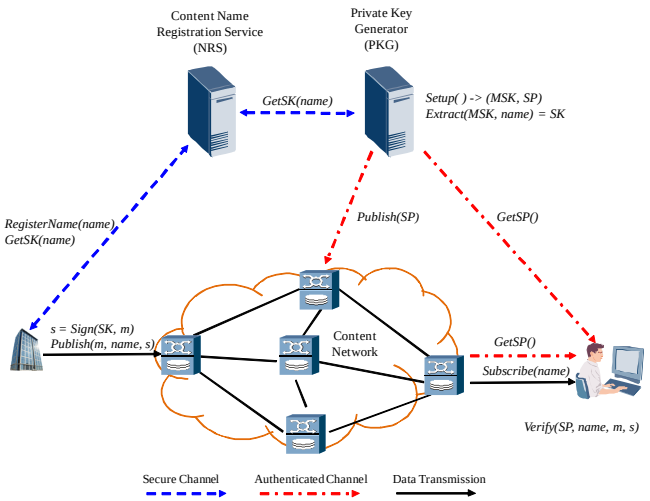

Fig. 1. System architecture and protocol.

be the name or prefix of a content; that is, the trust is built on top of the named object. With this, one assumption is that there is some authorization mechanism built in the network such that a dedicated name should be given to authorized publisher only. We believe this is reasonable, as it is the case in current Internet: DNS service binds domain names with IP addresses [1]. Furthermore, the desired authorization mechanism is not necessarily global, but can be discretionary for a domain or organization. For example, it is desired to have an authorization mechanism to enforce that a regular employee cannot publish content under the name and signature of her manager. We note that when a content is named with the identity of its provider, such as a smartphone device id or a domain name, these two approaches are the same.

Figure 1 illustrates our scheme from a high level view. The system consists of a trusted private key generator (PKG) that generates the system parameters (SP) and the master secret key (MSK). The PKG keeps MSK privately by itself, and publishes SP to the network. We assume user identity is available with existing trust infrastructure, such as email, phone number, or organizational identities.

With the first scheme, before a content provider publishes contents, it first obtains a secret (private) key (SK) with its identity from the PKG, which uses the `Extract` algorithm to generate the key with inputs of the identity and the MSK. Note that this is a one-time operation (dotted lines in Figure 1) for the content provider with a particular identity [2]. A secure channel is needed here for the content provider to obtain the SK. Optionally, the SK can be obtained in an offline manner, e.g., preloaded to a device when it is released or installed. After this, the provider can use the SK to sign contents that it publishes with `Sign` algorithm. Note that the identity of the

[1]We note that DNS-based attacks such as deny of service and cache poisoning do exist in the current Internet, which is orthogonal to the problem that we target in this paper.

[2]In general, one content provider may have multiple identities which are known to different consumers.

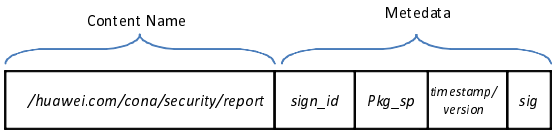

Fig. 2. Content metadata format.

provider is also included as metadata of the content. When a consumer receives a content from the network, it first obtains the SP of the PKG from network or through offline manner, and then uses `Verify` algorithm to verify the integrity of the content with the identity from the content metadata. Note that the operation of obtaining the SP is also a one-time operation for the consumer, which is different from that in the traditional PKI where the consumer needs to obtain individual publishers' certificates.

To leverage a content name as an identity, a name registration service (NRS) is introduced, as shown Figure 1. Before a content provider publishes a named content, it first registers the name with the NRS. If allowed, the NRS obtains the SK associated with the name from the PKG, and returns to the provider through a secure channel. The content is then signed by this SK. When received by a consumer, the content can be directly verified by its name with the SP. We note that although a malicious provider can intentionally publish content with a known-name, a consumer can detect this with failed signature verification, as it is not signed by the corresponding SK of the content name.

The format of content name and metadata is shown in Figure 2 and explained below.

- $content\_name$: human-readable content name;
- $sign\_id$ is the identity used for signature verification, and can be the identity of the content provider or the content name/prefix;
- pkg_sp is the SP of the PKG corresponding to the identity. This can be either the real SP, or the name of the SP that can be obtained from content network;
- sig is the signed hash of the content name, metadata, and data.

B. Protecting Confidentiality with IBE

One nice feature with IBE is that, in order to send a secret data, the sender does not need to obtain the public key certificate of the receiver, as that in the traditional PKI-based scheme. Instead, the sender can use the identity of the receiver to encrypt the data and publish it. Once received, the data can be decrypted by the receiver if it can obtain the SK from the PKG. If not, the data will not be decrypted. Thus, IBE enables very flexible data confidentiality protection. Especially for content network, where a content provider may not know potential receivers, thus cannot obtain their public key certificate in advance. Furthermore, a content name can also be an identity; that is, a provider can publish content with a name such that only a particular receiver can obtain the SK associated with it, e.g., via the name registration

service or pre-loaded in offline manner. With this, a content provider can ensure that only expected receivers can read the data, even without knowing them or obtaining their public key certificates. This is useful in many cases, such as, a content name can be selected by its content provider and the content can be distributed to network securely, even there is no pre-established secure communication channel.

Correspondingly, there can be two schemes for confidentiality protection with IBE: encrypting content with the identity of a receiver, or with a content name or its prefix. The processes are similar to that in IBS, except that the content provider uses `Encrypt` algorithm to encrypt data with identities, while the receiver uses `Decrypt` to obtain the cleartext of the content. In order to save computing cost for encryption and decryption for large contents, key encapsulation mechanism (KEM) can be used, i.e., a content can be encrypted with a (symmetric) data encryption key (DEK), which is then encrypted with the identity, such that crypto operations on content use the DEK which is usually much more efficient than public key algorithms. Like other public key cryptography algorithms, IBE and IBS can be used together. For example, a content can be encrypted with a receiver's identity, and then signed by its provider. For content shared within group of users, to leverage the caching capability of content network for content distribution, the content can be encrypted with a single shared group key, while the publisher controls the distribution of this common key to group users. Due to space limit we do not consider details of this.

C. A Hybrid Scheme

Similar to PKI, an IBC system relies on a centralized trusted party – the PKG. Obviously, a single centralize trusted party is not scalable, especially for Internet-scale network environment. Even worse, IBC usually works on the user or application level, which is very difficult to enforce all users or applications to trust a single entity.

Another challenge for an IBC system is related to secret key and system parameters distribution. As Figure 1 shows, for a typical IBC system, we need a secure channel for distributing the SK corresponding an identity, and an authenticated channel for obtaining the public SP of the PKG. For SK distribution, some out-of-band communication mechanisms can be used, such as pre-loading to a device or application, or offline handover, which are not scalable. For SP distribution, a basic requirement is that a user needs to verify its authenticity: that it really belongs to a particular PKG domain. Otherwise, the IBS or IBE may not work, as identity is bound with SK uniquely by the PKG. In general, online mechanism is needed for SP distribution, e.g., when a receiver does not have communication with a content provider before, it has to obtain the SP from network.

Towards a scalable solution, we further propose a hybrid scheme by integrating PKI and IBC. From a high level view, we do not consider PKG and NRS as global entities; instead, they can be a domain or organizational entity, which issues secret keys to local users. For example, an enterpirse can have its own PKG, which is the case in TrendMicro's identity-based email encryption solution for enterprise [1]. Secret key distribution can be built on existing authentication mechanisms within the enterprise, such as username/password or Kerberos.. In order to enable secure cross-domain communications, a user of domain A needs to obtain the SP of B in an authenticated way. For this purpose, we leverage PKI: the SP of domain A is signed by the the domain's PKI private key such that the user in domain B can verify the authenticity. Figure 3 shows the skeleton of CCN protocol, where Bob in domain B retrieves some content with Alice's name in domain A. The verification is done by 2-step verification: verifying the authenticity $DomA.PKG.SP$ with DomA's public key certificate, which is in turn verified by a certificate from a trusted CA.

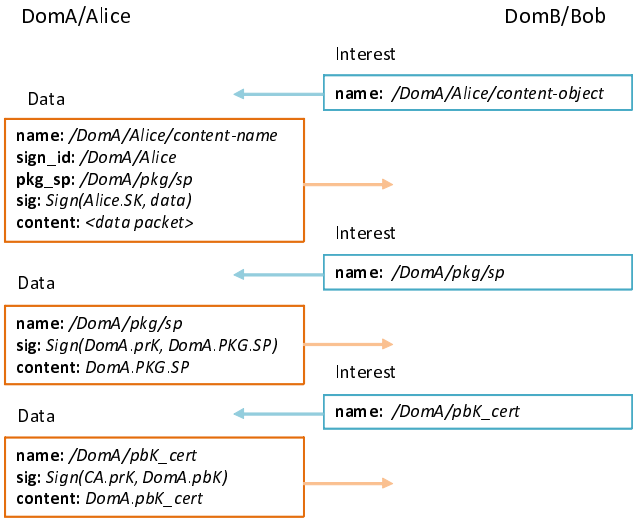

Fig. 3. Signature verification with PKI and IBS.

This scheme is scalable based on the fact that PKI is deployed in domain or AS-level in current Internet infrastructure. Particularly, DNSSEC, IPSec, and server-side SSL/TLS for web-based Internet services extensively rely on PKI-based trust infrastructure. In these protocols, public keys of domains are certified by a few trusted CAs, which has been evidenced as a scalable solution. We argue that PKI is viable for domain level trust management, while IBC is good for end user and device level trust management. In content network, requiring each content provider or consumer has a public key certificate is costly and key management becomes an obstacle for many applications.

Note that this hybrid scheme does not compromise the benefits for name-based trust and security with IBC. In particular, a consumer only needs a domain's public key certificate to verify the authenticity of a PKG's SP; after that, the trust is still built on top of trusting the identity of the content provider or content name, instead of individual user's public key certificate. Therefore certificate management burden is limited in domain level.

D. Discussion

Supporting flexible security policies: We expect variant security policies can be easily supported. First of all, flexible delegation can be achieved with identity-based signature, where different name prefixes can be used to sign the same content, e.g., with different levels of organizational identities, and a consumer can verify only with one identity (e.g., the complete one). Furthermore, a content receiver can delegate the decrypting capability to multiple users based on the content name, e.g., when she is absent [5]. With hierarchical identity-based encryption [6], a content can be decrypted with any user who owns an identity which is a substring of the target identity. For example, a content named `/huawei.com/cona/storage` can be decrypted by people with identity of `/huawei.com/cona` and `/huawei.com`, however it cannot be read by `/huawei.com/corenet` or `/huawei.com/cona/routing`. The NRS can enforce very flexible security policies on the content name that a provider can legitimately publish or a consumer can read.

User-level PKG: Similar to self-signed public key certificate, an individual user can act as a PKG, i.g., generates private keys by herself. This is useful when domain-based trust of PKGs is not viable, such as in P2P and social network applications [3]. User-level PKGs in content network can support very flexible trust management and security requirements, such as content names as identities that are trusted only within a social group of a user.

Comparing with PKI: We note that PKI-based certificate can also bind some user attributes with a public key, e.g., with X.509 extensions. However, we claim that there are fundamental difference from PKI-based approach and IBC for content oriented network. Specifically, with PKI, a user needs to obtain the public key certificate of the individual content publisher in order to verify the integrity and thus the trust of the content. While in IBS, a user leverages already known identities for the same purpose, and the public parameters (SP) of the PKG is common for all content publishers. Also, a content publisher doesn't need to pre-fetch the certificate of a receiver for secure communication with IBE. Instead, it can directly encrypt contents with the identity of an expected receiver or even the content name.

Key/Identity revocation: Certainly IBC-based solution is not a panacea for security. Actually, we believe there is no one-fit-all solution for security and trust in content network. Particularly, private (secret) key revocation has been a problem for IBC algorithms. Although several revocation mechanisms have been proposed in recent years [4], we believe it is an essential and more difficult problem than in PKI. It has been suggested in IBE algorithm [5] that private key expiration can be done with implementing ephemeral public identities. The public keys can be generated by including timestamp with identity depending on granularity, e.g. `bob@huawei.com || current-month`. The private key is then updated every month by the system in automatic way. This approach is used in commercial product [1]. Unlike traditional PKI, a user doesn't need to request a new certificate when a private key is revoked.

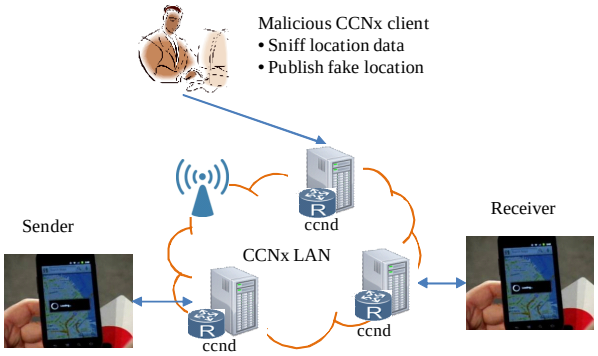

Fig. 4. Architecture of implementation prototype with Android application on CCNx network.

IV. Implementation

We have implemented our scheme with the CCNx project [3] on Android-based smartphone devices, and developed a Google Latitude-like [4] location-based application to share a user's location data to her friends in her contact list. A location content is named as `/ccnx/latitude/phone-number/loc`, where the device's 10 digit phone number is its identity. The data (concatenation of GPS coordinates) is encrypted with the receiver's (a friend in contact list) phone number, and then signed by the private key corresponding to the phone number of the sender. The PKG is implemented as a standalone application on a Linux box, which generates private keys based on phone numbers as inputs. We then manually load the private keys into the devices as a file stored in the phone SD card, which is then read by the application during runtime. The PKG's SP is named `/ccnx/latitude/pkg/sp` and published to the network by the PKG. The Android devices connect to the CCNx network via a WiFi access point. Figure 4 shows the overall architecture of our implementation prototype.

We ran the prototype and evaluation on Nexus S devices running Android 2.3. We use the open source Pairing-Based Cryptography (PBC) library [5] and implement one IBS [8] and one IBE [5] algorithms. The current IBE implementation is developed with the jPBC package, which is a Java porting of the PBC library written in C. When encrypting a message of 2048 bytes, the IBE encryption is almost 10 times slower than the RSA encryption. However, as the main time used in IBC is the pairing operations, which is data size independent, when working on large data size, the performance difference becomes smaller. For example, with IBE encryption to encrypt a 128-bit AES key and then use the AES key to encrypt a 5M bytes message, it takes around 19.526 seconds. For the same

[3] http://www.ccnx.org
[4] https://www.google.com/latitude
[5] http://crypto.stanford.edu/pbc

operation with a 1024-bit RSA key, it takes around 17.819 seconds. IBE decryption with AES decryption averages 19.117 seconds, with RSA decryption with AES decryption averages 17.419 seconds. We observed similar performance with IBS and RSA performance as the message size increases. The evaluation results are an average time with 30 measurements. We believe that when using native code implementing of IBC, such as with the Android Native Development Kit (NDK) [6], the performance will be much better than our current implementation.

V. RELATED WORK

Smetters and Jacobson [16] treat content authentication as a process of content self-certifying with authentication on linkage between names and content. This approach provides a ground secure content mechanism that can be a subsidy of our approach to ensure the integrity of names and content. Besides bootstrapping security for CCN network, VoCCN [10] demonstrates the feasibility to obtain security protection from the existing VoIP architecture without extra effort, when transfer it into CCN. Reusing existing security-enhanced mechanism to achieve security requirement by CCN is also a future direction for our research. NDN proposes to use SPKI/SDSI with local namespaces for trust management, where local authorities are used to build trust for content. PGP uses a web of trust model to derive the trust of a certificate. Usually, a content consumer needs to obtain a chain of certificates in order to verify the authenticity of a content, which we conjuncture the key management cost and runtime performance will be a concern for many applications on mobile platforms.

Some ICN networks use hash of content or public keys for both of content names (identifiers) and integrity verification purposes, such as NetInf [7], PSIRP [11], and DONA [12]. Although theoretically IBC can work with these, it is not desirable as one of our main motivations is to derive intuitive and direct trust from content names, instead of solely integrity verification. That is, hierarchical and user-readable names in CCN and NDN have more benefits with our approach. We also note that although we aim to achieve security from content names, our scheme does not violate the principle of separating authenticator from identifiers and locator [16], as the integrity verification is still performed with a signature, instead of hash of content. Uniquely, our scheme binds readable content name and signature seamlessly.

IBC has bee proposed for network security since Shamir's first IBC proposal on email systems in 1984. Appenzeller and Lynn [2] have employed IBC to design a network layer security protocol. In [13], an identity-based key agreement protocol has been introduced for the network layer. In the context of mobile network, an identity-based key agreement system for mobile telephony in GSM and UMTS network has been implemented in [17]. Similar to our hybrid scheme, to tackle the scalability issue of large scale PKI, Smetters and Durfee [15] use IBC for email and IPSec protocols, where domain level parameters are bound with DNS. Our work is the first attempt to apply IBC in ICN for trust and security.

[6]http://developer.android.com/sdk/ndk/index.html

VI. CONCLUDING REMARKS

We propose an identity-based signature and encryption mechanism for integrity and trust verification and confidentiality protection of data in recently proposed content oriented network such as CCN and NDN. As transportation in content network is based on named content instead of communication channel, our solution uniquely leverage the name as identity to bootstrap content-based trust. For sake of scalability we propose a hybrid solution by combining PKI and IBC. We have implemented a prototype of our scheme and an application with CCNx on Android devices and local network environment to demonstrate the effectiveness of integrity verification and confidentiality protection.

REFERENCES

[1] The true costs of e-mail encryption: Trend micro ibe (identity-based) vs. pki encryption, http://us.trendmicro.com/imperia/md/content/us/pdf/products/enterprise/emailencryption/the_true_cost_of_email_encryption_6-2010.pdf, Oct. 2010.

[2] G. Appenzeller and B. Lynn. Minimal-overhead ip security using identity based encryption. Technical report, Voltage Inc., 2002.

[3] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: An online social network with user-defined privacy. In *Proc. of SIGCOMM*, 2009.

[4] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *Proc. of ACM CCS*, 2010.

[5] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proce. of CRYPTO*, 2001.

[6] D. Boneh, E. Goh, and X. Boyen. Hierarchical identity based encryption with constant size ciphertext. In *Proc. of Eurocrypt, LNCS 3493*, 2005.

[7] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren. Secure naming for a network of information. In *Proc. of IEEE Global Internet Symposium*, 2010.

[8] F. Hess. Effiecent identity based signature schemes based on pairings. In *Proc. of ACM SAC*, 2002.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proc. of ACM CoNEXT*, 2009.

[10] V. Jacobsonn, D. K. Smetterss, N. Briggs, M. Plass, and P. Stewart. Voccn: voice over content-centric networks. In *Proc. of ACM Workshop on Re-architecting the Internet*, 2009.

[11] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, and C. Esteve. Lipsin: line speed publish/subscribe inter-networking. In *Proc. of ACM SIGCOMM*, 2009.

[12] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proc. of ACM SIGCOMM*, 2007.

[13] C. Schridde, M. Smith, and B. Freisleben. An identity-based key agreement protocol for the network layer. In *Security and Cryptography for Networks*, volume 5229 of *LNCS*, pages 409–422. 2008.

[14] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO*, 1985.

[15] D. K. Smetters and G. Durfee. Domain-based authentication of identity-based cryptosystems for secure email and ipsec. In *Proc. of Usenix Security Symposium*, 2003.

[16] D. K. Smetters and V. Jacobson. Securing network content. Technical report, PARC, 2009.

[17] M. Smith and et al. Securing mobile phone calls with identity-based cryptography. In *Advances in Information Security and Assurance*, volume 5576 of *LNCS*, pages 210–222. 2009.

[18] L. Zhang and et al. Named data networking (ndn) project. Technical Report NDN-0001, PARC, 2010.