Decentralized Name-based Security for Content Distribution using Blockchains

Nikos Fotiou* and George C. Polyzos*,#

*Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
Evelpidon 47A, 113 62 Athens, Greece
{fotiou, polyzos}@aueb.gr

#Department of Computer Science and Engineering
Jacobs School of Engineering
University of California, San Diego
La Jolla, CA 92093-0443, USA
polyzos@cs.ucsd.edu

***Abstract*—User, content, and device *names* as a security primitive have been an attractive approach especially in the context of Information-Centric Networking (ICN) architectures. We leverage Hierarchical Identity Based Encryption (HIBE) to build (content) name-based security mechanisms used for securely distributing content. In contrast to similar approaches, in our system each user maintains his own *Private Key Generator* used for generating the *master secret key* and the public *system parameters* required by the HIBE algorithm. This way our system does not suffer from the key escrow problem, which is inherent in many similar solutions. In order to disseminate the system parameters of a content owner in a fully distributed way, we use *blockchains*, a distributed, community managed, global list of transactions.**

I. Introduction

Information-Centric Networking (ICN) is an emerging networking paradigm that has received attention recently by the research community (e.g., see [1] for a survey on ICN research). ICN architectures use content names as the main ingredient of their (inter-)networking functions. Therefore, it comes as a natural choice to consider content names[1] as the basic security primitive for ICN. Using content names as the main building block of security mechanisms offers some intriguing advantages. Firstly, content names can be human readable, therefore, they can be memorable (as oppoesd for example to RSA public keys), thus it should be easier to disseminate them using out of band mechanism, e.g., by printing them on a business card, or including them in a slide presentation. Secondly, content names can be predictable, therefore, it could be easy to predict the name of a content item that has not yet been created, e.g., the name of the next chunk of a live video stream. Lastly, content names can be hierarchical, reflecting real world organization and business relationships.

In this paper we are concerned with content distribution in ICN networks (although our solution is generic enough and can also be used in other similar architectures). In particular, we consider the case in which a *content owner* wants to share content with some *subscribers*. We wish to provide *content integrity protection* and *content provenance verification* based on content names. Content integrity protection is an integral part of any content distribution system, since it assures that a (content) item has not been modified during transmission. In order to highlight the advantages of (content) name-based integrity protection consider the example of an item being made available (i.e., published) by several endpoints. Using legacy content integrity mechanisms, either all these endpoints should share the same public/private key pair, which raises security concerns, or a subscriber should learn the public key of the endpoint from which she received a (content) item.

In our system all these entities would share some publicly available system parameters, as well as, a content-specific secret. A subscriber that knows the system parameters can verify a digital signature over the item no matter the providing endpoint. Content provenance verification allows a subscriber to verify that an endpoint that hosts some content has been authorized by the content owner to do so. This property, which is implemented using a controlled *content storage delegation* algorithm, is useful in cases where a subscriber wants to receive an item only from endpoints trusted by the content owner, e.g., for accounting reasons, spam prevention, phishing protection, etc. Our work here does not aim to provide content confidentiality and access control, nevertheless, content confidentiality and access control solutions can be easily used in conjunction with our approach and system. Content storage delegation provides a secure way for a content owner to authorize a third party to host the content.

In order to achieve our goals we leverage our previous work on name-based security and trust presented in [3]. That paper defines mechanisms that take advantage of Hierarchical Identity Based Encryption (HIBE) [4]. HIBE is a public key encryption scheme in which an identity can be used as a public key. Our system uses content names as HIBE public keys. HIBE specifies an entity, namely the *Private Key Generator* (PKG), which generates the private keys that correspond to each identity. All HIBE algorithms require as input some publicly available *System Parameters*, SP, which are PKG specific. A single system-wide PKG results in a *key escrow* problem, since the PKG knows all private keys, whereas multiple PKGs would require a resolution mechanism that

[1] In this paper we use the terms names, identifiers and identities interchangeably (see, e.g., https://en.wikipedia.org/wiki/Identifier for a discussion). Naming in ICN is analayzed in [2].

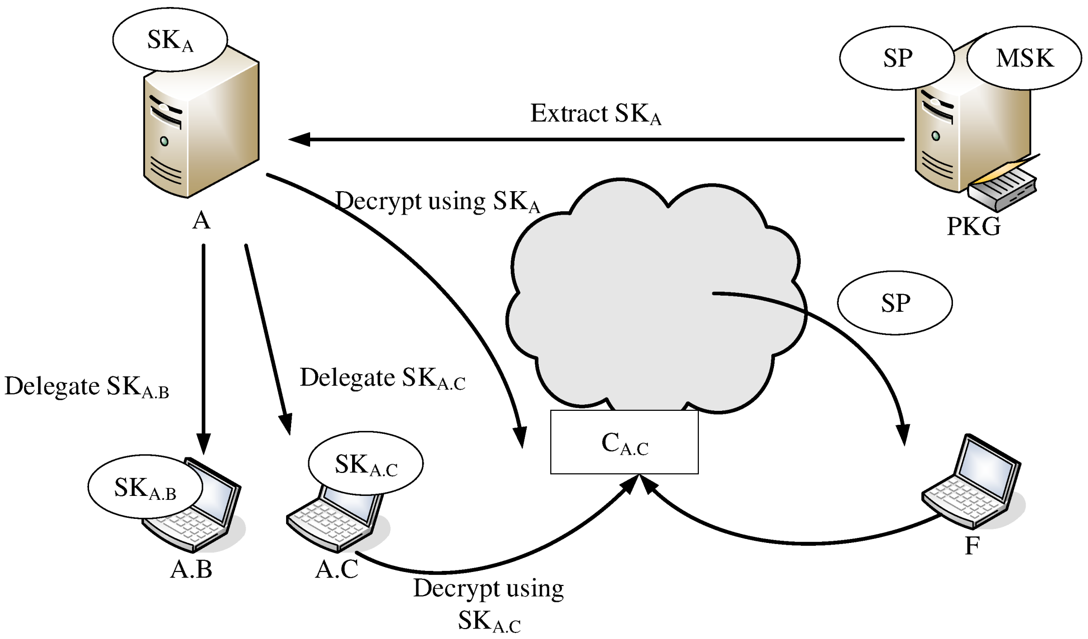

Fig. 1. HIBE algorithms. The PKG generates a MSK and SP, and makes SP publicly available. The PKG extracts the secret key for identity A. The owner of the identity A delegates the secret keys that corresponds to identities $A.B$ and $A.C$ to the corresponding owners. An entity creates a ciphertext using the identity $A.C$ as the public key. Both the owner of the identity A and the owner of the identity $A.C$ are able to decrypt this ciphertext.

maps identities to SP. In our system we consider one PKG per content owner, eliminating this way the key escrow problem, and we use a *blockchain* to disseminate SP. Blockchains are data structures that securely record *transactions* and are maintained by a distributed network of trustless nodes. Our implementation uses the blockchain provided by *Namecoin*[2], an open source information registration and transfer system based on the Bitcoin cryptocurrency.

The remainder of this paper is organized as follows. Section II briefly introduces HIBE and blockchains. Section III presents the design of our solution. We evaluate our solution in Section IV. Finally we present related work in Section V and we conclude our paper in Section VI.

II. Background

A. Hierarchical Identity-based Encryption

An Identity Based Encryption (IBE) scheme is a public key encryption scheme in which an identity (or a name, i.e., an arbitrary string) can be used as a public key. An IBE scheme is specified by the four algorithms, `Setup`, `Extract`, `Encrypt` and `Decrypt`, summarized as follows:

- `Setup` is executed by a Private Key Generator (PKG). It takes as input a security parameter k and returns a `master-secret key` (MSK) and some `system parameters` (SP). The MSK is kept secret by the PKG, whereas the SP are made publicly available.
- `Extract` is executed by a PKG. It takes as input SP, MSK, and an identity ID, and returns a `secret key` SK_{ID}.
- `Encrypt` takes as input an identity ID, a message M, and SP, and returns a ciphertext C_{ID}.
- `Decrypt` takes as input C_{ID}, the corresponding private decryption key SK_{ID}, and returns the message M

HIBE schemes consider hierarchical identities and specify an additional algorithm, `Delegate`:

- `Delegate` takes as input SP, SK_{ID_1}, and an identity $ID_1.ID_2$ and outputs $SK_{ID_1.ID_2}$

The `Delegate` algorithm is of particular importance as it enables the owner of an identity A to generate SKs for other identities that use A as a prefix, without communicating with the PKG. Fig. 1 illustrates the HIBE algorithms.

B. Blockchains

A blockchain is a *distributed ledger* of transactions maintained by a network of trustless nodes. Each block of the blockchain contains a list of transactions organized in a Merkle tree. New blocks are added to the blockchain by the *miners*. The addition of a new block involves the computation of a solution to a computationally intensive puzzle. The miner that successfully solves the puzzle floods the block in the network: if this block becomes accepted by at least 51% of the miners, then it is added in the blockchain. Miners have incentives (usually monetary) to calculate a valid block. Any network node can participate in the network of miners. The

[2] https://namecoin.info/

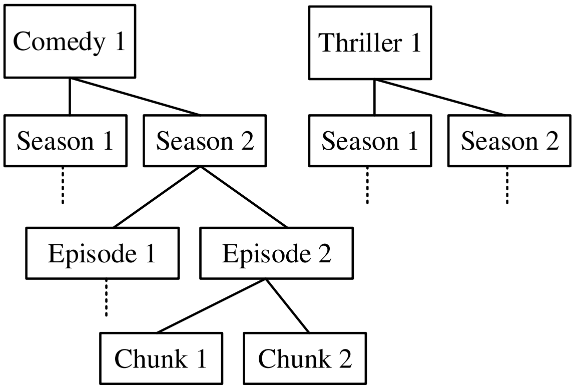

Fig. 2. Name space organization

most well known blockchain is the one used by the Bitcoin crypto currency.[3]

Blockchains are often referred to as a *democratic* way of maintaining transactions as they rely on consensus for confirming transactions and require no central authority.

III. Design

A. Setup

Our design assumes globally unique content names (at least in the scope of a specific application). The granularity and the semantics of the names are application specific.

Content name management can be facilitated by organizing names in direct acyclic graphs (DAGs). In that case, only the root of each graph should be globally unique. In order to illustrate this concept we discuss in the following the use case of a TV studio. Suppose that a TV studio is the content owner of two TV series, namely 'Comedy 1' and 'Thriller 1'. The namespace in that case could be organized as in Fig. 2. In this figure there are two DAGs, one for each series. The root of each graph is the name of the series. Each series is composed of 'seasons', each season of 'episodes' and each episode of 'chunks'. This information organization is reflected in the DAGs. Depending on the application, various levels of content granularity can be considered, therefore, it could be possible for a user to request a content item named 'Comedy 1.Season 2' corresponding to a whole season of the 'Comedy 1' series, or it may be possible to request a content item named 'Comedy 1.Season 2.Episode 2. Chunk 1' corresponding to a specific content item chunk.

Each content owner generates the (public) SP required by the HIBE algorithm using his own PKG, as well as, a (secret) SK for each content name. Moreover, content owners *register* all globally unique names in the blockchain, including in the registration message the SP. The registration process is implemented as a new transaction in which the owner associates a content name with–among other things–his SP. It should be noted that SP are content owner unique and not content name unique, i.e., the registration transactions of content names belonging to the same content owner will include the same SP. Therefore, in the previous use case, the studio, will register in the blockchain the names 'Comedy 1' and 'Thriller 1', including in each registration message the same SP (Fig. 3).

The particular blockchain implementations may provide various safeguards in addition to name uniqueness guarantee. These safeguards, depending on the application, may include protection of trademarks, removal of content names, etc.

B. Content storage delegation

Each piece of content is stored in a content *storage node*. A storage node may belong to a content owner or it may be *authorized* by the content owner to store some content items, or even a portion of the content name space (e.g., it may host all episodes of 'Season 1' of 'Comedy 1'). A content owner authorizes a content storage node to store content on his behalf by using the *Trust Delegation* algorithm specified in [3]. In a nutshell, the content owner executes the HIBE *Delegate* algorithm and generates the SKs (secret keys) that correspond to the names of the content items (or the portion of the name space) the authorized node will store, and distributes those keys to this node. It should be noted at this point, that the rightful owner of these keys is the content owner and that the storage node simply acts on the owners behalf, therefore, the fact that the content owner knows these keys is not considered key escrow. Moreover, key distribution should be secured (however key distribution is out of the scope of this paper).

Back to our TV studio example, suppose that the TV studio wants to authorize 'CDN A' to store all episodes of 'Season 1' of 'Comedy 1', it generates $SK_{Comedy1.Season1}$ and securely distributes it to 'CDN A'. Note that 'CDN A' is now able to generate $SK_{Comedy1.Season1.Episode1}$, $SK_{Comedy1.Season1.Episode1.Chunk1}$ and so forth, using the HIBE *Delegate* algorithm.

C. Content retrieval

A subscriber that wishes to (securely) retrieve a piece of content has first to learn the SP that correspond to the content owner by *querying* the blockchain. This query should include the content name of the desired content item, or the root of the DAG in which the content name in question belongs. For example, if a subscriber is interested in receiving 'Comedy 1.Season 2.Episode 2.Chunk 1' she should query the blockchain for the SP of the owner of 'Comedy 1'. Note that the root of each DAG is globally unique, therefore the subscriber does not know any owner specific information. Moreover, SP are content owner wide, therefore, if the same subscriber is interested in 'Thriller 1.Season 2', provided that she knows that these two pieces of content belong to the same owner, she does not have to query the blockchain again. As a next step, the subscriber issues a standard ICN *content retrieval* request, which is routed to an appropriate storage node by the underlay ICN network. The corresponding response may include a digital signature that can be used to verify the

[3] https://blockchain.info/

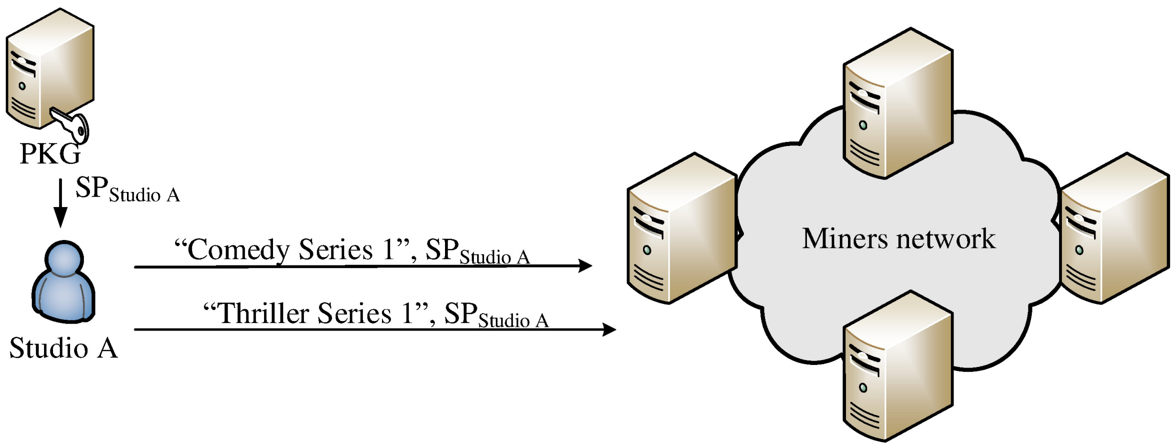
PKG
$\mathrm{SP}_{\mathrm{Studio\ A}}$
Studio A
"Comedy Series 1", $\mathrm{SP}_{\mathrm{Studio\ A}}$
"Thriller Series 1", $\mathrm{SP}_{\mathrm{Studio\ A}}$
Miners network

Fig. 3. Name registration

integrity of the received content. This signature is generated by using the following digital signature procedure [3]:

1) Digital signature: Assuming that the underlay HIBE algorithm is CCA secure, a digital signature scheme can be trivially constructed using the following two algorithms [5]:

- `Sign`: takes as input SP, a message M, a SK_{NAME}, and a secure hash function H, and outputs a digital signature $Sign_M = SK_{NAME.H(M)}$. The digital signature $Sign_M$ is constructed by using the `Delegate` algorithm of the HIBE scheme with input SP, SK_{NAME}, $NAME.H(M)$.
- `Verify`: takes as input the SP, H, M, a digital signature $Sign_M$ and the $NAME$ of the signer. Then:
  1) Selects a random number r.
  2) Encrypts r using the HIBE `Encrypt` algorithm with input $NAME.H(M)$, r, SP and produces a ciphertext C.
  3) Verifies that C can be decrypted using the HIBE `Decrypt` algorithm, with input C, $Sign_M$, SP.

Only the entity that owns SK_{NAME} is able to generate $Sign_M$. Moreover, since $Sign_M = SK_{NAME.H(M)}$ Step 3 of the verification algorithm is successful iff the digital signature is valid.

A subscriber can also verify content *provenance*, i.e., that a piece of content is stored by an authorized node, by using the following provenance verification procedure:

2) Provenance verification: A subscriber $S1$ is able to verify that a node $N1$ is authorized to host a piece of content named $Content1$ owned by an owner with SP $SP1$ by using the following challenge response protocol:

1) $S1$ selects a random number r, executes the HIBE `Encrypt` algorithm with input $Conent1$, r, $SP1$ and sends the resulting ciphertext C to $N1$.
2) $N1$ uses the HIBE `Decrypt` algorithm with input C, $SK_{Content1}$, $SP1$ and sends the output back to $S1$.
3) $S1$ verifies that the received response equals to r.

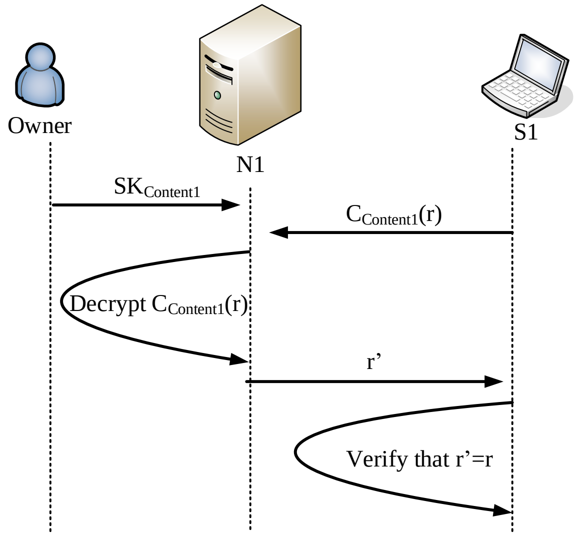
Owner
N1
S1
$\mathrm{SK}_{\mathrm{Content1}}$
$\mathrm{C}_{\mathrm{Content1}}\mathrm{(r)}$
Decrypt $\mathrm{C}_{\mathrm{Content1}}\mathrm{(r)}$
r'
Verify that r'=r

Fig. 4. Content provenance verification

If the verification of the last step succeeds, then it means that $S1$ knows $SK_{Content1}$, therefore, it is authorized to host 'Content1'. This protocol is illustrated in Fig. 4.

D. Extensions

We now describe some extensions that can be considered for our system.

1) Private content retrieval: A subscriber is able to privately request a piece of content, named 'Content1' using the following algorithm:

1) Select a symmetric encryption key K.
2) Encrypt the content item request using K and generate $Enc(request)$.
3) Encrypt K, using the HIBE `Encrypt` algorithm with input 'Content1'.
4) Flood the network with $C_{Content1}(K), Enc(request)$.

Only nodes that know $SK_{Content1}$ should be able to decrypt $C_{Content1}(K)$ and therefore $Enc(request)$. The symmetric encryption K can also be used for encrypting the content item, providing this way content confidentiality.[4]

2) Content authentication: Content authentication assures that the retrieved content is what the subscriber asked for, i.e., content authentication provides a mapping between the content name and the content data. Content authentication can be achieved in our system by registering in the blockchain the output of a hash function applied over the data of a content item. This registration should include the complete name of the item in question. Similarly, a subscriber should be able to query the blockchain using the complete item name and retrieve the item hash.

IV. Evaluation

A. System implementation and performance evaluation

In our evaluation we used the Lewko-Waters HIBE scheme for prime order settings [6] implemented with the Charm-Crypto library[5][7]. In [3] we evaluated the performance of this HIBE implementation and we showed that it is practical. As blockchain we used *Namecoin*. Namecoin is an open source information registration and transfer system based on the Bitcoin cryptocurrency. The blocks of the Namecoin's blockchain can be calculated by Bitcoin miners, therefore, there is already a critical mass of miners.

The Namecoin blockchain allows name registration, as well as, the association of some data with a name. Currently, Namecoin limits the size of this data to 520 bytes. The SP produced by the Lewko-Waters HIBE scheme are larger in size. As a consequence, in our implementation, an owner registers the *hash* of his SP. For this reason, a subscriber should learn, using out of band mechanisms, the actual value of the SP. This can be achieved either by including in the blockchain transaction a URL where the SP are located, or by including the SP in the first content transmission. In either case, the subscribers should calculate the hash of the received SP and compare the output of the hash function with the value in the blockchain. Subscribers are able to query the Namecoin blockchain using many existing libraries, such as *nmcontrol*[6]. The Namecoin software downloads the complete blockchain and updates it periodically. Currently, the size of the blockchain is almost 1.4 GB and querying it for a record requires less than 5 ms. Alternatively, the blockchain can be downloaded by a trusted entity which can then respond to requests transmitted over a secured channel. For example, namecoin-core[7] provides a REST interface that allows entities to perform namecoin resolution requests over HTTP.

B. Security evaluation

The Lewko-Waters HIBE scheme is Chosen-Ciphertext Attack (CCA) secure, therefore content provenance authentication and digital signature algorithms are secure. An interesting security problem concerns node *de-authorization*, i.e., how can a content owner remove the authorization from a storage node to store some content. If the content owner does not interact with many nodes, he can simply update his SP and make sure that subscribers update *cached* SP frequently enough. This of course requires interaction with the blockchain and may come with a monetary cost.[8] Another solution is to use *key expiration*. For example, suppose that an owner wants to delegate the storage of content item $Content1$ to storage node $N1$ and that the delegation should expire on the 30^{th} of April, 2019. The owner can generate $SK_{Content1\#20190430}$, i.e., a secret key that corresponds to the content name with appended the expiration date, and store this key in $N1$. $N1$ should now use this key in all algorithms and should include the key expiration date in all responses.

Another interesting security problem is key revocation, since the loss of a SK means that requests to the associated content name can be hijacked, therefore it should be revoked. The key revocation mechanism described in [3] can be used to mitigate this issue. This mechanism specifies that each content item should have two names: a name that identifies the item and a name that is used as a public key. The latter name is constructed by appending to the former a *serial number*. Every time a new SK is required the serial number is incremented. In order to learn the current serial number of an item name the following solutions can be applied: (i) use out of band mechanisms, (ii) resolve the serial number using the blockchain, or (iii) have the communicating endpoints agree on a serial number calculation algorithm (e.g., use as serial number the current date).

The use of blockchains contributes to the security of our system. In blockchains there is no single point of failure as, for example, in Web PKI, where a single certificate authority can jeopardize the security of TLS [8]. Decisions in blockchains are based on consensus and, as long as at least 51% of miners behave honestly, a blockchain is secure.

V. Related work

Smetters and Jacobson [9] use a resolution service that maps a content name to a set of security information items, including the public keys of authorized publishers in order to provide various security properties to the content itself.

Zhang et al. [10] utilize the identity-based encryption (IBE) scheme proposed by Boneh and Franklin [5] and the identity-based signature scheme proposed by Hess [11] in order to provide name-based security and trust mechanisms for the NDN architecture [12]. They use a legacy PKI system in order to deliver the necessary system parameters. Our system uses HIBE, which offers some significant advantages compared to IBE. Moreover, our system uses a blockchain to deliver system parameters, alleviating the need for a PKI.

Mahadevan et al. [13] propose a key resolution mechanism for the CCN architecture. Their mechanism can be used to map a content name to (among other things) the content

[4] Note that this protocol does not provide forward secrecy. For a key exchange protocol that provides forward secrecy refer to the *Authenticated key exchange* construction presented in [3].

[5] Our source code is available at: https://github.com/nikosft/HIBE_LW11

[6] https://github.com/namecoin/nmcontrol

[7] https://github.com/namecoin/namecoin-core

[8] This is true in the case of Namecoin.

owner's public key. This mechanism can be used instead of the blockchain in order to map a content name to the content owner's SP. Nevertheless, it requires global roots of trust in order to bootstrap and it is CCN-specific. In contrast, blockchains do not require any global root of trust and they are not bound to any particular architecture.

Similarly, Yu et al. [14] have developed a *trust schema* for NDN that provides content consumers a way to discover which keys to use in order to verify digital signatures over the data. Our system does not require such a schema since the key used for generating a digital signature is the name of the signed content itself.

Various other research efforts specify content encryption mechanisms in order to provide content confidentiality and/or access control for ICN (e.g., [15], [16], [17]). Our system is orthogonal to these systems since it was not designed to and does not provide content confidentiality and access control. Nevertheless, it can be used in conjunction with all such mechanisms in order to provides these additional properties.

Our solution uses a blockchain to deliver SP. Alternative approaches use the name resolution infrastructure to deliver TLS keys. E.g., DANE TLSA [18] uses DNNSEC. Some researchers argue that these approaches suffer from security risks, since the name resolution infrastructure is (usually) controlled by government bodies. Blockchains do not suffer form these problems since any user is allowed to participate in a blockchain.

VI. Conclusions and future work

We presented decentralized name-based security mechanisms that aim to secure content distribution in ICN and similar architectures. Our mechanisms leverage Hierarchical Identity Based Encryption (HIBE) to provide *content storage delegation*, *content provenance verification*, and *content integrity protection*. Our solution does not suffer from the key escrow problem, which is inherent in many other designs. Moreover, our solution uses blockchains to deliver the Systems Parameters, SP. Blockchains do not rely on any central authority, or on pre-trusted nodes, and provide interesting security properties.

Our scheme is generic enough and can be incorporated into various ICN architectures, or any other similar system. In our prototype implementation we used separate, IP based, software, provided by Namecoin, in order to interact with the blockchain. Future work in this area could include the investigation of an ICN based blockchain implementation. Moreover, in our implementation we used the Lewko-Waters HIBE scheme. The advantage of this scheme is that it allows identifier hierarchies of arbitrary depth with constant-size SP. However, this comes at the cost of having SP with size larger than the size supported by the corresponding field of Namecoin, which we used. In order to solve this problem, either alternative blockchain implementations, or alternative HIBE schemes could be explored in the future.

Acknowledgment

Supported in part by H2020 ICT POINT and by AUEB-RC.

References

[1] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[2] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in Content-oriented Architectures," in *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*, ser. ICN '11. ACM, August 2011.

[3] N. Fotiou and G. C. Polyzos, "Enabling NAME-Based Security and Trust," in *Trust Management IX*, ser. IFIP Advances in Information and Communication Technology, C. D. Jensen, S. Marsh, T. Dimitrakos, and Y. Murayama, Eds. Springer International Publishing, 2015, vol. 454, pp. 47–59.

[4] A. Lewko and B. Waters, "Unbounded HIBE and Attribute-Based Encryption," in *Advances in Cryptology - EUROCRYPT 2011*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6632, pp. 547–567.

[5] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in *Advances in Cryptology - CRYPTO 2001*, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Springer Berlin Heidelberg, 2001, vol. 2139, pp. 213–229.

[6] A. Lewko, "Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting," in *Advances in Cryptology EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Springer Berlin Heidelberg, 2012, vol. 7237, pp. 318–335.

[7] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[8] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *Proceedings of the 2013 Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 291–304.

[9] D. Smetters and V. Jacobson, "Securing Network Content," PARC, Tech. Rep., 2009.

[10] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, and G. Wang, "Towards name-based trust and security for content-centric network," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, October 2011, pp. 1–6.

[11] F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science, K. Nyberg and H. Heys, Eds. Springer Berlin Heidelberg, 2003, vol. 2595, pp. 310–324.

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[13] P. Mahadevan, E. Uzun, S. Sevilla, and J. Garcia-Luna-Aceves, "CCN-KRS: A Key Resolution Service for CCN," in *Proceedings of the 1st International Conference on Information-Centric Networking*, ser. ICN'14. New York, NY, USA: ACM, 2014, pp. 97–106.

[14] Y. Yu, A. Afanasyev, D. Clark, k. claffy, V. Jacobson, and L. Zhang, "Schematizing Trust in Named Data Networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*, ser. ICN '15. New York, NY, USA: ACM, 2015, pp. 177–186.

[15] J. Kuriharay, E. Uzun, and C. Wood, "An encryption-based access control framework for content-centric networking," in *IFIP Networking Conference (IFIP Networking), 2015*, May 2015, pp. 1–9.

[16] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-Based Access Control for Content Centric Networks," in *Proceedings of the 2Nd International Conference on Information-Centric Networking*, ser. ICN'15. New York, NY, USA: ACM, 2015, pp. 147–156.

[17] M. Mangili, F. Martignon, and S. Parabosch, "A cache-aware mechanism to enforce confidentiality, trackability and access policy evolution in Content-Centric Networks ," *Computer Networks*, vol. 76, pp. 126–145, 2015.

[18] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," IETF, RFC 4033, 2005.