Enabling NAME-based security and trust

Nikos Fotiou and George C. Polyzos

Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
Patision 76, 104 34, Athens, Greece
{fotiou,polyzos}@aueb.gr

Abstract. An integral component of almost any security and trust system is endpoint identity verification. The predominant identification primitive, used in most contemporary systems, is the digital certificate. A digital certificate binds a NAME (i.e., an "official way to refer to an entity") to a cryptographic public key, which is then used for the NAME verification. In this paper, we propose a NAME verification system that does not rely on digital certificates. Our solution uses Hierarchical Identity Based Encryption (HIBE) to allow fine-grained NAME verification, trust delegation and attribute-based access control. For the delivery of the necessary system parameters we propose an approach that leverages the NAME registration and resolution systems, eliminating the need for a Public-Key Infrastructure. As proof of concept, we implement and evaluate our system using the Lewko-Waters HIBE scheme and DANE-DNSSEC.

1 Introduction

Almost every entity in the Internet has at least one *NAME*, i.e., *an official way to indicate an entity uniquely* [11]. Examples of NAMEs are domain names, e-mail addresses, and electronic product codes. NAMEs can be bound to a cryptographic public key using a *Digital Certificate* (DC); DCs can then be used for NAME verification. This process is an essential component of many security and trust systems. In this paper, we postulate that security and trust systems can be built directly on NAMEs without relying on DCs. What is more, we argue that the NAME hierarchy can be used to implement trust delegation and access control mechanisms. To this end, we propose a solution in which NAMEs hold the role of public keys. In the following use case scenario we illuminate some of the advantages of the use of NAMEs as public keys.

Service A enables decentralized content sharing. Users of this service are registered using a short, memorable nickname. The organizers of *Conference B* use *Service A* in order to allow conference attendants to exchange files. A sponsor of *Conference B* has prepared an electronic gift card and it has encrypted it with the public key "Service A.Conference B.attendant". The gift card and the decryption key are "transmitted" to the conference attendants during the "welcome session". During the

conference, a presenter wishes to share her slides with the audience. She includes her nickname "nickname A" in her first slide and broadcasts a list of files and their location, digitally signed with the private key that corresponds to "Service A.nickname A". Moreover, the presenter has delegated the NAME "Service A.nickname A.presentation.live" to a video streaming service which is now authorized to stream her presentation.
Various features of NAME-based security and trust systems can be identified in this use case: it is possible to create a ciphertext using a public key (NAME) that will be generated in the future, NAMEs can be small and memorable and they can even be included in a presentation slide or a business card, NAME-based digital signatures can be easily verified, simply by using the NAME of the signer, and NAMEs can be structured and sub-NAMEs can be delegated to third parties, enabling them to act on behalf of the NAME owner.
In this paper we propose a solution that enables NAME-based security and trust systems. We take advantage of the structure of NAMEs and we design constructions for trust delegation and attribute-based access control. Our system uses Hierarchical Identity Based Encryption (HIBE). HIBE is selected since, compared to plain IBE, it facilitates (private) key generation and transitivity of trust. In our system, HIBE system parameters can be disseminated using the name resolution infrastructure. As a proof of concept we implement our system using the Lewko-Waters HIBE scheme [10].
This paper is organized as follows: In Section 2 we discuss related work in this area. In Section 3 we briefly present HIBE. In Section 4 we detail our construction, whereas in Section 5 we present its implementation and evaluation. Finally our conclusions are presented in Section 6.

2 Related work

Related work in this area mostly concerns Identity Based Encryption (IBE). Despite using NAMEs as keys, IBE, is not as flexible as HIBE. The generation of a private key always involves communication with a third party and trust delegation is not straightforward. Moreover IBE schemes cannot be used for generating digital signatures and an additional Identity Based Signature (IBS) scheme is required in systems that use IBE.
Smetters and Durfee [13] utilized IBE encryption to provide secure email delivery and encrypted network traffic. In their scheme they used DNSSEC to deliver the system parameters (we detail these parameters in Section 3). Smart [12] used IBE in order to implement authenticated key agreement, whereas Green and Giuseppe [5] utilized IBE to implement proxy re-encryption (i.e., transformation of a ciphertext encrypted with key A to a ciphertext encrypted with key B). All these works consider a particular application of IBE, whereas our work proposes a holistic NAME-based trust enabler system.
The work of Zhang et al. [14] has very similar goals with our system. In their work, Zhang et al., utilized the IBE scheme proposed by Boneh and Franklin [3] and the IBS scheme proposed by Hess [6] in order to

provide name-based security trust mechanisms for the NDN Information Centric Networking (ICN) architecture [7]. Moreover they used a legacy PKI in order to deliver system parameters. Our work improves [14] in the following: (a) we utilize HIBE, therefore, (i) private key generation is faster, (ii) no separate signature scheme is required, and (iii) trust can be delegated and (b) we use the name resolution infrastructure to deliver system parameters which, as we argue, offers better fault isolation and easier security breach detection.

3 Background

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

- `Setup`: it 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 SP are made publicly available.
- `Extract`: it 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 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 $SK_{ID_1.ID_2}$

`Delegate` algorithm is of particular importance, as it enables the owner of in identity to generate SKs for its descendants in the identity hierarchy without the involvement of the PKG.[1] As a consequence, a message encrypted using an identity ID as the public key, can be decrypted by any of the ancestors of ID in the identity hierarchy. Fig. 1 illustrates the main components and algorithms of HIBE.

Recent advances in HIBE have led to practical schemes, such as the solution proposed by Lewko and Waters [10]. This scheme supports arbitrary number of identities, it does not require the identity hierarchy depth to be known during `Setup`, and it has constant size SP.

4 System design

Our system assumes that every *administrative domain* is identified by a NAME and maintains its own PKG (we discuss in Section 5.3 the granularity of administrative domains). All entities in an administrative domain are hierarchically organized. The position of an entity in the domain hierarchy is reflected in its NAME.

[1] On the contrary, key generation in IBE schemes always involves the PKG.

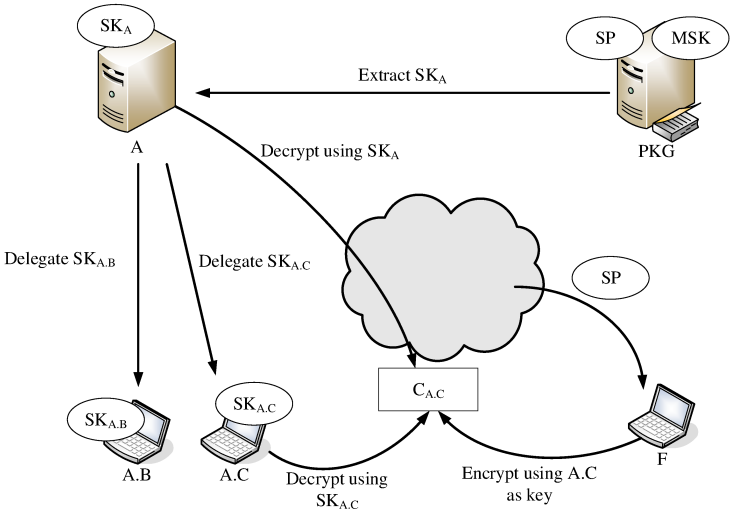

Fig. 1. HIBE overview

The NAME of an entity is used as a public key. The PKG generates the SKs of the "first level" entities, using the `Extract` algorithm, the "first level" entities generate the SKs of the "second level" entities, using the `Delegate` algorithm, and so forth. In the following we present some constructions that can be used to build security and trust systems.

4.1 Basic constructions

Being public key encryption based, our construction supports the following constructions:

Digital signature A digital signature over a piece of content authenticates the identity of the signer and protects content integrity. Assuming that the underlay HIBE algorithm is CCA secure a digital signature scheme can be trivially constructed using the following two algorithms [3]:

- `Sign`: takes as input SP, a message M, a SK_{NAME}, 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 SP, H, a message 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.

Authenticated key exchange An authenticated key exchange protocol enables two parties to authenticate themselves and to establish a secure communication channel. In the following we describe a Diffie-Hellman (D-H) based authentication key exchange protocol between two entities with NAMEs N_1 and N_2. For the sake of simplicity we assume that both entities use the same SP. Let g, p be the public parameters of the D-H protocol, then:

1. N_1 selects a random number r_1, computes $u_1 = g^{r_1} \pmod p$, signs it using `Sign` algorithm–described previously–and sends u_1 and $Sign_{u_1}$ to N_2.
2. N_2 selects a random number r_2, computes $u_2 = g^{r_2} \pmod p$, signs it using `Sign` algorithm, and sends u_2 and $Sign_{u_2}$ to N_1.
3. Both users verify the signatures and if the verification is successful they compute $u = g^{r_1 * r_2} \pmod p$ which is used as the shared secret key

4.2 Additional constructions

The use of hierarchical NAMEs and HIBE in our system enables some additional constructions.

Trust delegation Suppose an entity with NAME N_1 that wants to use a content distribution network CDN_A to disseminate some files. Our system enables (i) CDN_A to digitally sign stored files on behalf of N_1, and (ii) users to verify that CDN_A has indeed been authorized by N_1 to store these files. This is achieved using the following process.
Let $N_1.files$ be the prefix of the NAME of the files. N_1 executes `Delegate` algorithm and generates $SK_{N_1.files}$ which is securely transmitted to CDN_A. CDN_A can digitally sign a file on behalf of N_1 using the `Sign` algorithm described previously, with input SP, the file, and $SK_{N_1.files}$.
A user U_1 can "challenge" CDN_A to prove that it has been indeed authorized by N_1 to host $N_1.files$ using the following procedure:

1. U_1 selects a random number r_1 and sends it to CDN_A
2. CDN_A computes $SK_{N_1.files.r_1}$, using the `Delegate` algorithm and sends the result back to U_1.
3. U_1 selects a random number r_2, encrypts r_2 using the HIBE `Encrypt` algorithm with input $N_1.files.r_1$, r_2, SP and produces a ciphertext C
4. U_1 Verifies if C can be decrypted using the HIBE `Decrypt` algorithm, with input C, $SK_{N_1.files.r_1}$, SP

Attribute-based access control Using HIBE it is possible to encrypt data in a way that only certain categories of users can decrypt it. Consider the example of a laboratory where the head of the lab should be able to decrypt data encrypted for lab members, but not vice versa. Moreover, each lab member should be able to decrypt only data encrypted for her. In this scenario the head of lab should be equipped with a SK that corresponds to the NAME of the lab (e.g., SK_{lab}), whereas lab members should have a SK that corresponds to a NAME prefixed with the lab NAME (e.g., $SK_{lab.member01}$). Data encrypted for *lab.member01* can be decrypted by both the head of the lab and member01. Moreover data encrypted for *lab* can only be decrypted by the head of the lab. Another interesting example is the case of a spam communication detection filter. By revealing to the filter the key SK_{lab}, it should be able to decrypt and inspect all messages, whereas users will be able to decrypt only their own.

4.3 Delivery of system parameters

A crucial aspect of our system (and of any (H)IBE based scheme) is SP delivery. One solution that can be considered the use of the name resolution service. In this subsection we describe such an approach without binding it to a particular name resolution system. In the next section we detail our DNSSEC-based implementation.

It is assumed that the administrative domain NAMEs are of a hierarchical form. Moreover, it is assumed that these NAMEs are "registered" to a naming *registration* system, composed of reliable "brokers" which are also organized using the same hierarchical form. The root brokers are responsible for managing the root of the NAME space, the first level brokers are responsible for managing the first level of the NAME space and so forth. Every broker has a self-generated public/private key pair. The public keys of the root brokers are considered well-known and trusted, whereas the public keys of the rest of the brokers are digitally signed by their direct ancestor, i.e., the public keys of the first level brokers are signed by a root broker, the public keys of the second level brokers are signed by a first level broker, and so forth. The NAME of an administrative domain is registered to a leaf broker (the *registrar*). Each administrative domain NAME is associated with a public/private key pair. The public part of this key is signed by the NAME registrar, whereas the private part of this key is used for digitally signing SP.

We now distinguish two forms of NAME *resolution*: (i) the case in which the NAME resolution system is coupled to the naming registration system, and (ii) the case in which these systems are decoupled. The first case is the most commonly used (e.g., in DNS). In this case a name resolution request follows the brokers hierarchy and "collects" signed public keys and signed responses . For example, the NAME resolution of *.gr.edu.mmlab* will result in the collection of the public key of the broker that manages the *.gr.edu* NAME space, signed by a root broker and the public key of the broker that manages the *.gr.edu.mmlab* NAME space, signed by the *.gr.edu* broker. Since the public keys of the root brokers are well known, this chain of signatures can be trivially verified. Fig. 2

illustrates this case. The second case of naming resolution is used in

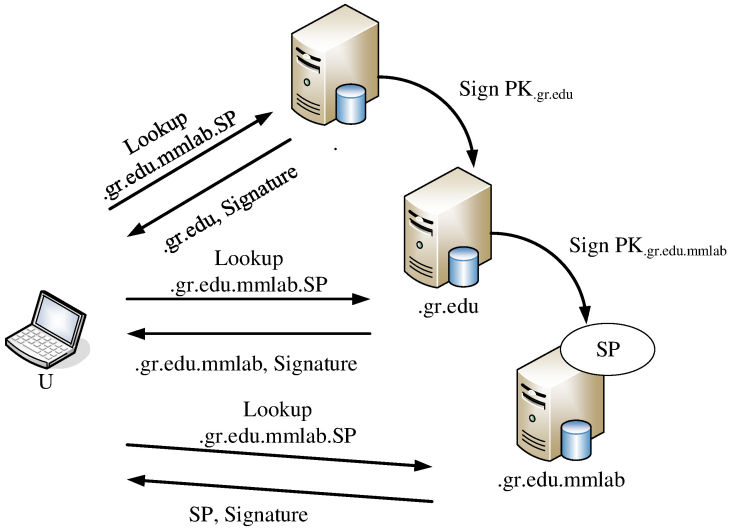

Fig. 2. Coupled NAME registration and NAME resolution systems

contemporary architectures, such as ICN architectures, that either use other forms of naming resolution systems, e.g., DHT name resolutions systems–such as in [8], or they do not use a naming resolution system at all (e.g., [7] floods–in a controlled–way the NAMEs in the network). In this case the complete chain of signatures of the NAME registration system should be "advertised". Fig. 3 illustrates this case.

4.4 Key revocation

The loss of a SK means that the associated NAME can be hijacked, therefore it should be revoked. In order to prevent this event, our systems considers the usage of two NAMEs per entity: a NAME that identifies the entity and a NAME that is used as a public key of that entity. The latter NAME is constructed by appending to the former a *serial number* (e.g., the public key of `lab.user01` can be `lab.user01-0034` with 0034 being the serial number). Every time a new SK is required the serial number is incremented. In order to learn the current serial number of a NAME the following solutions can be applied: (i) use out of band mechanism (e.g., in the use case discussed in the Introduction the serial number could have been included in the first slide), (ii) resolve the

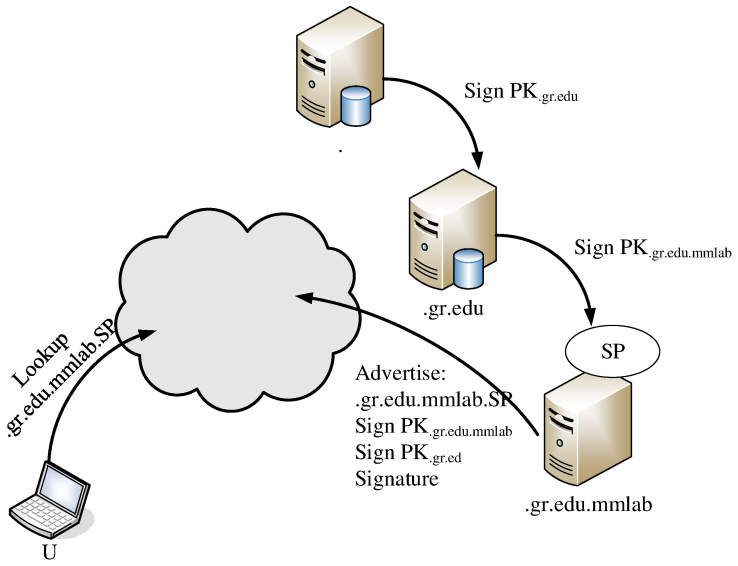

Fig. 3. Decoupled NAME registration and NAME resolution systems

serial number using the name resolution service (e.g., perform a NAME lookup for `lab.user01.SN`), (iii) have the communicating endpoints to agree out-of-band for a serial number (e.g., use as a serial number the current date). These solutions can be used in combination by applying each of them at different hierarchy levels.

An interesting application of dual NAMEs is *key expiration*, i.e., the ability to construct keys with certain lifetime. Supposed that it is desirable to create keys that are valid only for the current month: by creating SKs of the form $SK_{NAME||Current\_Month}$, where $||$ denotes concatenation, and by enforcing users to use $NAME||Current\_Month$ as the public key, the desired functionality can be achieved.

5 Evaluation

As a proof of concept we have implemented our system using the Lewko-Waters [10] HIBE scheme. The Lewko-Waters scheme is fully secure and it is based on bilinear maps applied over the elements of a group G of order p, where p is a prime number.[2] In our implementation G is a subset of the elements of a supersingular Elliptic Curve (EC). In this setup, public keys are elements of $\mathbb{Z}_p$, messages are elements of G (i.e., they are points of an EC), and ciphertexts are elements of $\mathbb{Z}_{p^a}$, where

[2] We have considered modification of the scheme for prime order settings [9]

a is a small number affected by the selection of G. The security of this scheme is based on the hardness of the Discrete Logarithm problem in $\mathbb{Z}_{p^a}$.

As a NAME registration and resolution service we consider DNSSEC. In DNSSEC all DNS servers are equipped with a public/private key pair. The private key is used for digitally signing DNS records. Moreover, a digest of the public key of a DNS server is stored as a signed record by its ancestor DNS server. Finally, all DNS-clients are pre-configured with the public keys of the root DNS servers. Following an approach similar to DANE TLSA [2], our implementation stores SP as a KEY record in the DNS zone of the administrative domain, using the alias SP. Therefore, supposing that the NAME of a domain is `.gr.edu.mmlab`, the resolution of the NAME `.gr.edu.mmlab.SP`, will result in the secure transmission of SP.

5.1 Performance evaluation

The Lewko-Waters scheme for prime order settings has been implemented using the Charm-Crypto tool [1] in Python 2.7.[3] All measurements have been performed using an Ubuntu 12.04 PC, equipped with an Intel i5 processor and 2GB RAM.

For our evaluation we have used a supersingular EC of order 512. The elements of this curve are mapped to $\mathbb{Z}_{p^2}$, i.e., $\mathbb{Z}_{1024}$. The size of the base64 encoding[4] of the SP for this particular setup is 5816 bytes. It should be reminded that the size of SP remains constant and that the scheme does not require any information regarding NAMEs in order to create SP.

One of the drawbacks of the Lewko-Waters scheme is that SK length and ciphertext size, as well as, the encryption and decryption times, are affected by the level of the NAME that is used as the public key. Table 1 shows the size of SK, the encryption time of a random number $r \in G$ (the hash of which can be used as they key to a symmetric encryption scheme) and the size of the ciphertext C_{NAME}, as a function of the level of the NAME used as the public key.

NAME level	SK size	Enc. time	C_{NAME} size
0	6340	32	1172
1	7236	63	2068
2	8128	84	2964
3	9024	103	3848
4	9916	122	4744

Table 1. SK size in bytes, Encryption time in ms and ciphertext size in bytes, as a function of NAME level

[3] Source code available at: https://github.com/nikosft/HIBE_LW11

[4] We are using base64 since SP are stored as a DNS record

The decryption time of a ciphertext C_{NAME} does not depend on the level of the NAME used as the public key, it only depends on the level of the NAME that corresponds to the SK that is used for the decryption. As an example the decryption time using $SK_{lab.user}$ is the same for $C_{lab.user}$ and $C_{lab.user.file1}$. Table 2 shows the decryption time in ms, for decrypting a ciphertext C generated using a NAME of 4^{th} level, as a function of the level of the NAME of the decrypting entity.

NAME level	Dec. Time
0	47
1	83
2	119
3	153
4	186

Table 2. Decryption time in ms as a function of NAME level

The execution time of the `Delegate` algorithm also depends on the level of the NAME of the entity that performs the delegation. Table 3 shows the execution time of the `Delegate` algorithm measured in ms, as a function of the NAME level of entity that performs the delegation.

NAME level	Deleg. Time
0	60
1	105
2	132
3	165

Table 3. Execution time of the `Delegate` algorithm, measured in ms

It should be noted here that in all algorithms the hash of a NAME is used (and not the NAME itself), therefore the length of a NAME does not affect the algorithms performance.
As it can be observed from the above results, the overhead introduced by our scheme it totally acceptable.

5.2 Security evaluation

The security of the basic constructions of our system depends on the security of the underlay HIBE algorithm. Currently there are fully secure algorithms that can be safely used in our system.
When our system is used and providing that the SP that correspond to an endpoint NAME are known, it is possible to establish a secure communication channel with that endpoint without any additional information. This is a big improvement compared to legacy certificate based

schemes. It should be noted here that knowing the SP that correspond to a NAME is not equivalent to knowing a security certificate, since SP are administrative domain wide. Therefore, by learning once the SP of an administrative domain, one can establish secure communication channels with any of the entities that belong to this domain

The usage of the name resolution system for disseminating SP makes Man in the middle attacks harder, compared to Web PKI. Supposedly, a malicious entity wants to impersonate an entity with NAME *.gr.edu.mmlab*. This malicious entity should persuade the brokers that manage the *.gr.edu* NAME space (i) to sign a fake public key, and (ii) to redirect NAME lookup requests to a fake broker.[5] In the current DNSSEC system there is a single such entity. On the contrary, in Web PKI every certificate authority (CA) can lawfully issue a certificate for any entity, even without the entity's consent: if the CA is considered trusted, then this certificate is successfully validated. A recent study [4] found that, when it comes to Web PKI, most end-points blindly trust 683 CAs. Each of these CA can issue a valid certificate for *any* entity.

Another advantage of the usage of the name resolution system for disseminating SP is that security breaches have local effects and it is easier to detect them: Supposedly, a malicious user succeeds in luring a broker that manages the *.gr.edu* NAME space into generating fake public keys and giving fake responses. These public keys can only be used for attacking NAMEs that use *.gr.edu* as a prefix. In contrast, in Web PKI a malfunctioning CA may affect an entity with which it has no direct relationship whatsoever. Moreover, in our system, an entity can periodically probe the name resolution system in order to proactively prevent security attacks.

Another point of consideration is the security risk introduced by name "registrars". Indeed, it is a widespread concern that with DNSSEC a malicious registrar may alter the public key of a domain. This is a valid concern, but it should be clarified that if a NAME is "locked" to a particular registrar, the NAME owner should only rely on the trustworthiness of this particular registrar (in contrast to Web PKI where a domain owner should rely on the trustworthiness of any pre-trusted CA). Moreover, it should be clarified that the NAME registrar never learns the private key(s) of an administrative domain.

5.3 Discussion

An important aspect of our scheme is the granularity of an administrative domain. The granularity of an administrative domain is determined mainly based on two factors: (i) the private key escrow problem (inherent in any (H)IBE scheme) which enables a PKG to decrypt ciphertexts and (ii) the ability of an entity to decrypt all messages encrypted with the NAME of any of its successors. Administrative domain granularity affects the depth of the NAME hierarchy (which in return affects the overhead introduced by HIBE), as well as, the number of NAMEs that

[5] We assume that the entities that manage the root name space cannot be "lured".

use the same SP. We can consider various levels of granularity for an administrative domain, ranging from very low (e.g., a whole multinational company) to very high (e.g., a user that maintains his own domain, therefore his own PKG). It should be noted that even when the highest level of granularity is considered, our scheme has many advantages compared to an IBE-based scheme as, even in this case, users still can digitally sign files and delegate trust without communicating with the PKG.

6 Conclusions

In this paper we designed a solution that enables NAME-based security and trust systems, using the Lewko-Waters [10] HIBE scheme. Our system achieves trust delegation and access control, enabling new applications, such as secure content delegation. Our system considers the NAME-resolution infrastructure for delivering the system parameters which offers significant security advantages. Our implementation shows that this scheme is feasible and practical, since the overhead that it introduces is acceptable. We believe that many emerging (inter)networking architectures, including Information-Centric Networking (ICN) and the Internet of Things (IoT), can benefit from the adoption of our scheme in their design. The leverage of the role of NAMEs offers content owners new possibilities and allows the construction of new forms of trust relationships. Of course, as we demonstrated through our implementation, our solution can also be applicable to existing communication systems. We envision our solution being used as an alternative to "ad-hoc" solutions that try to solve the problems of Web PKI (e.g., certificate pinning), as well as, as an alternative to existing secure communication applications (e.g., PGP).

Future work in this domain includes implementation of our scheme for various kinds of name resolution systems and its incorporation into new architectures. We will also explore the possibilities of embedding our scheme in smaller devices, in order to provide NAME-based trust for the IoT. Moreover, in this work we considered that administrative domains are isolated from each other. Of course this is not always the case: there can be administrative domains within other domains or there might be administrative domains that have some form of trust relationship. These cases are an exciting field for applying contemporary cryptographic solutions that are based on (H)IBE.

Acknowledgment

This research has been co-financed by the European Union (European Social FundESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)Research Funding Program: Aristeia II/I-CAN.

References

1. Akinyele, J.A., Garman, C., Miers, I., Pagano, M.W., Rushanan, M., Green, M., Rubin, A.D.: Charm: a framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering **3**(2), 111–128 (2013)
2. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: DNS security introduction and requirement. RFC 4033, IETF (2005)
3. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: J. Kilian (ed.) Advances in Cryptology - CRYPTO 2001, *Lecture Notes in Computer Science*, vol. 2139, pp. 213–229. Springer Berlin Heidelberg (2001)
4. Durumeric, Z., Kasten, J., Bailey, M., Halderman, J.A.: Analysis of the https certificate ecosystem. In: Proceedings of the 2013 Conference on Internet Measurement Conference, IMC '13, pp. 291–304. ACM, New York, NY, USA (2013)
5. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: J. Katz, M. Yung (eds.) Applied Cryptography and Network Security, *Lecture Notes in Computer Science*, vol. 4521, pp. 288–306. Springer Berlin Heidelberg (2007)
6. Hess, F.: Efficient identity based signature schemes based on pairings. In: K. Nyberg, H. Heys (eds.) Selected Areas in Cryptography, *Lecture Notes in Computer Science*, vol. 2595, pp. 310–324. Springer Berlin Heidelberg (2003)
7. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking Named Content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp. 1–12. ACM, New York, NY, USA (2009)
8. Katsaros, K.V., Fotiou, N., Vasilakos, X., Ververidis, C.N., Tsilopoulos, C., Xylomenos, G., Polyzos, G.C.: On inter-domain name resolution for Information-Centric Networks. In: R. Bestak, L. Kencl, L. Li, J. Widmer, H. Yin (eds.) NETWORKING 2012, *Lecture Notes in Computer Science*, vol. 7289, pp. 13–26. Springer Berlin Heidelberg (2012)
9. Lewko, A.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: D. Pointcheval, T. Johansson (eds.) Advances in Cryptology EUROCRYPT 2012, *Lecture Notes in Computer Science*, vol. 7237, pp. 318–335. Springer Berlin Heidelberg (2012)
10. Lewko, A., Waters, B.: Unbounded HIBE and Attribute-Based Encryption. In: Advances in Cryptology - EUROCRYPT 2011, *Lecture Notes in Computer Science*, vol. 6632, pp. 547–567. Springer Berlin Heidelberg (2011)
11. Rackoff, C.: On "identities", "names", "NAMES", "ROLES" and Security: A Manifesto. Tech. Rep. 2011/214, IACR Cryptology ePrint Archive (2011)
12. Smart, N.: Identity-based authenticated key agreement protocol based on weil pairing. Electronics Letters **38**, 630–632(2) (2002)

13. Smetters, D.K., Durfee, G.: Domain-based administration of identity-based cryptosystems for secure email and ipsec. In: Network Protocols (ICNP), 2011 19th IEEE International Conference on (2003)
14. Zhang, X., Chang, K., Xiong, H., Wen, Y., Shi, G., Wang, G.: Towards name-based trust and security for content-centric network. In: Network Protocols (ICNP), 2011 19th IEEE International Conference on, pp. 1–6 (2011)