Journal of Information Security and Applications 36 (2017) 20–31

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Analysis of hierarchical identity based encryption schemes and its applicability to computing environments

Renu Mary Daniel*, Elijah Blessing Rajsingh, Salaja Silas

Department of Computer Sciences Technology, Karunya University, Tamil Nadu, India, 641114

ARTICLE INFO

Article history:
Available online 2 August 2017

Keywords:
Hierarchical identity based encryption
Bilinear pairings
Prime order groups
Composite order groups
Performance analysis

ABSTRACT

Hierarchical Identity Based Encryption (HIBE) enhances the scalability of Identity based encryption scheme, by sharing the workload of the root Private Key Generator (PKG) among multiple lower-level PKGs, facilitating intermediate key escrows and private key delegation. Owing to its structure, HIBE can be deployed to provide access control in cloud, pervasive computing systems, wireless sensor networks and Massively Multiplayer Online Role-Playing Games (MMORPGs). Additionally, HIBE can be used to perform search on encrypted data, forward secure encryption, fully private communication, limited delegation and damage control. This paper evaluates different approaches in the construction of HIBE protocols to determine practical frameworks. Specific criterions like cryptographic proof models, tightness of the reduction, recipient anonymity, hardness assumptions, bounded depth, revocability, types of pairing and ciphertext indistinguishability properties, were used as benchmarks for assessing each scheme. The efficiency in terms of storage and computation overhead, was estimated to identify suitable protocols for securing different computing environments. The future prospective applications of HIBE protocols were also investigated.

1. Introduction

Identity Based Encryption (IBE) system is a simplified, certificate-free Public Key Infrastructure (PKI) model. In IBE, a user's public key can be derived directly from a well-known identity string, like e-mail id or social security number. The corresponding private key is generated by the Private Key Generator (PKG) from a secret master key. The PKG also generates the system public parameters required for encryption and key generation. The notion of IBE was introduced by Shamir [56]. Boneh and Franklin [8] devised the first practical IBE scheme in the Random Oracle Model (ROM), using Weil or Tate pairings on elliptic curves. On a general elliptic curve, the discrete logarithm problem is as difficult to break as in a generic cyclic group, due to the absence of subexponential discrete log algorithms. Hence, a 160 bit elliptic curve provides equivalent security of a 1024 bit finite field [39]. As IBE is based on bilinear pairings on elliptic curves, it provides better security at smaller key sizes.

The PKG in an IBE system is responsible for user authentication, private key extraction and generation of system parameters, for all communicating entities. In a larger network, a more scalable access control solution with load balancing is desirable. Inspired by the hierarchical structure of the certificate authorities in PKI, Horwitz and Lynn [28] introduced the concept of hierarchical identity based encryption, along with the formal security definitions using a two-level HIBE. At the top of the hierarchy, the root PKG generates system public parameters and a master secret key. The root PKG then generates private keys for domain PKGs at the lower level. A domain PKG does not generate any system parameters, but can create its own master secret key. The domain PKGs are responsible for user authentication and private key generation, in their respective domains. The public key of each user will be a tuple consisting of the user identity appended to the public key of its parent entity. Such a hierarchical structure provides load balancing, damage control and resilience.

The rest of the paper is organized as follows. Section 2 describes the basic HIBE protocol along with the security definitions. Section 3 documents a brief discussion about the factors influencing the design of a practical HIBE scheme. Section 4 details on the different approaches in the construction of HIBE schemes along with the strengths and weaknesses of each model. Section 5 provides an insight into the properties of revocable HIBE protocols. Section 6 depicts an extensive theoretical analysis of the performance of different HIBE schemes based on storage and communication overheads. Section 7 provides an account of the applications

* Corresponding author.
E-mail addresses: renudnl@gmail.com (R.M. Daniel), elijahblessing@gmail.com (E.B. Rajsingh), blessingsalaja@gmail.com (S. Silas).

http://dx.doi.org/10.1016/j.jisa.2017.07.005
2214-2126/© 2017 Elsevier Ltd. All rights reserved.

of HIBE. Based on the performance analysis, suitable HIBE schemes for securing different computing environments, were identified. Section 8 provides the concluding remarks.

2. Basic HIBE system

This section provides a description of the basic HIBE protocol along with the security definitions. A HIBE protocol is comprised of five algorithms, i.e., initial root set-up, key extraction, delegation, encryption and decryption.

Root set-up* ($\boldsymbol{\lambda}$)*: During set-up phase, the root PKG takes a security parameter λ, and computes the system public parameters *PP* and generates master secret key *mk*.

Key extraction (I, PP, mk): The key extraction algorithm uses the master key *mk* and public parameters *PP*, to compute the private key for identity vector $\vec{I}$ at depth *j*, provided, $j \leq L$, where, *L* is the maximum hierarchy depth. The private key for $\vec{I}$ is denoted as $k_{\vec{I}}$.

Delegation ($\vec{I}, PP, I', k_{\vec{I}}$): An identity $\vec{I}$ at depth *j*, generates the private key for identity I' at depth $j+1$, using public parameters *PP* and its private key $k_{\vec{I}}$, provided, $j+1 \leq L$, where, *L* is the maximum depth of the hierarchy. The public key for I' is denoted as $\vec{I}: I'$ and the private key is $k_{\vec{I}:I'}$.

Encryption (M, PP, $\vec{I}$): To encrypt a message intended for a recipient with identity vector $\vec{I}$, the sender uses public parameters *PP* and the public key $\vec{I}$ of the recipient, to generate the ciphertext *C*.

Decryption (C, PP, $k_{\vec{I}}$): The intended recipient with identity vector $\vec{I}$, decrypts the ciphertext *C*, using the public parameters *PP* and its private key $k_{\vec{I}}$, to obtain the message *M*.

The security game between the reduction algorithm and the probabilistic polynomial time adversary, is as follows:

Set-up: The challenger runs the set-up algorithm to generate the system public parameters *PP* and a master secret key *mk*. It keeps *mk* to itself and gives *PP* to the adversary.

Query Phase: The adversary issues queries $q_1, \ldots, q_m$, where, each q_i is either a key extraction query or a decryption query.

Key extraction Query: The adversary requests the private key of identity vector $\vec{I}$ at depth *j*, provided, $j \leq L$, where *L* is the maximum hierarchy depth. The challenger issues the private key $k_{\vec{I}}$ to the adversary.

Decryption Query: The adversary requests the decryption of ciphertext *C* corresponding to identity vector $\vec{I}$ at depth *j*, ($j \leq L$). The challenger decrypts *C* and returns *M* to the adversary.

Challenge phase: The adversary submits two equal length messages $\{M_0, M_1\}$ and the challenge identity vector $\vec{I}^*$ at depth j^*, $j^* \leq L$. The challenger randomly chooses message M_b, where, $b \in \{0, 1\}$ and sends the ciphertext C^*= *Encryption* (M_b, PP, $\vec{I}^*$) to the adversary.

The adversary again repeats the query phase by adaptively issuing more private key queries and decryption queries, under the constraint that the private key of $\vec{I}^*$ or any of its ancestor identities cannot be queried. The decryption oracle cannot be queried for the decryption of C^*.

Guess Phase: The adversary finally outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

The HIBE scheme is IND-ID-CCA secure if the advantage of the adversary in winning the above security game is negligible, for the given security parameter λ. If the adversary is restricted to issuing only private key queries in the above game, the system is said to be IND-ID-CPA secure. For an anonymous HIBE scheme, during the guess phase the adversary must submit two challenge identities $\{\vec{I}_0^*, \vec{I}_1^*\}$ along with the equal length messages $\{M_0, M_1\}$. The challenger randomly chooses $b \in \{0, 1\}$ and sends the ciphertext $C^* =$ *Encryption* (M_b, PP, $\vec{I}_b^*$) to the adversary. During the guess phase, the adversary wins the game if it correctly determines *b*, with non-negligible probability.

3. Factors influencing the design of a practical HIBE protocol

This section offers an insight into various aspects that influence the design and practical applicability of a HIBE protocol. Short discussions about the relevance of bilinear pairings, ciphertext indistinguishability models, tightness of a reduction, constant ciphertext size, recipient anonymity, hardness assumptions, ciphertext observability, revocability and bounded depth in the construction of HIBE is provided.

3.1. Types of pairings

Bilinear pairings are efficiently computable, non-degenerate maps between two elliptic curve sub-groups G_1 and G_2, of the same prime order, to a multiplicative subgroup G_T, of a finite field. A function $\hat{e}: G_1 \times G_2 \rightarrow G_T$ is called a bilinear map, if it satisfies the properties of bilinearity, non-degeneracy and efficient computability. Pairings are important in HIBE schemes, since it determines the size of the keys, system parameters and ciphertext. There are mainly two kinds of pairings used in cryptography: symmetric and asymmetric. Due to the ease of representation, symmetric pairing: $G_1 \times G_1 \rightarrow G_T$, is the most sought after construct, but is less flexible and least efficient [21]. Asymmetric pairing is of the form $\hat{e}: G_1 \times G_2 \rightarrow G_T$. A pairing is of type 1, if there exists an efficiently computable homomorphism ϕ from G_2 to G_1, as well as its inverse ϕ^{-1} from G_1 to G_2. Symmetric pairings hence fall in type 1. Type 2 consist of asymmetric pairings, where the homomorphism ϕ from G_2 to G_1 exists, but ϕ^{-1}does not exist. Type 3 also consist of asymmetric pairings, but in this case, neither ϕ, nor ϕ^{-1} is efficiently computable. It has been proven that, the representation size and computational cost of type 3 pairings are lower than in type 2 pairing [14]. The most practical HIBE schemes are constructed in Type 3 setting, as the computation and storage overheads are minimal [34]. In the asymmetric setting, group G_1 has least representation size, hence, usually, both public parameters and ciphertext will be elements of G_1 and private keys are represented in G_2 [45].

3.2. Adaptive security

According to the security definitions formulated by Horwitz and Lynn, any adversary that queries the key generation oracle for the private key corresponding to a particular identity, can compute the private keys of all the descendants of that identity by running the delegation algorithm. Obviously, if the prefix of the challenge identity has already been queried, then the security game becomes trivial. It is desirable to prove the security of a HIBE protocol in the adaptive security model (IND-ID-CPA/IND-ID-CCA) [24,28], where, the adversary can issue any finite number of decryption or key generation queries adaptively before and after submitting the challenge identity, with the restriction that the challenge identity or any of its prefixes did not appear in the query phase. Canetti et al. proposed a weaker notion of security called selective security (IND-sID-CPA/IND-sID-CCA) [11,12], where, the adversary must reveal the challenge identity even before the system setup phase. Evidently, this allows the challenger to partition the identity space appropriately, to avoid abort conditions in [24] and obtain a tight reduction. Later Boneh and Boyen [5] observed that any selectively secure IBE scheme can be converted to a fully secure system, with an inefficient security reduction O(2^n), where, n is the bit length of the identity. Whereas, for a HIBE scheme, the security degradation is O(2^{nL}), L being the maximum depth of the hierarchy. Hence, it is desirable to construct HIBE protocols with full adaptive security.