Broadcast Encryption with Multiple Trust Centers and Dynamic Coalitions

Christopher A. Seaman[1] Kent D. Boklan[2] Alexander W. Dent[3]

***Abstract*—We demonstrate an extension of hierarchical identity-based encryption (HIBE) from the domain of a single Trusted Authority (TA) to a coalition of multiple independent Trusted Authorities with their own hierarchies. Coalitions formed under such schemes may be temporary or dynamic in membership without compromising security. In this paper we give an instantiation with formal security model and a proof of security against selective-identity chosen-plaintext attacks in the standard model based upon the difficulty of solving the Bilinear Decisional Diffie-Hellman (BDDH) problem.**

I. Introduction and Motivation

A. Identity-Based Encryption

Identity-based encryption (IBE) [Sha84] has become an attractive alternative to public-key encryption schemes. A Public Key Infrastructure (PKI) allows for encryption only after obtaining a certified copy of the recipient's public key. In identity-based encryption, a message is encrypted using a personal identifier such as an email address rather than using a certified public key obtained from a PKI. The public key infrastructure is replaced by an identification system and a single public set of common parameters.

Identity-based encryption is most often considered in the context of one-to-one communication within a single Trusted Authority (TA). This TA is the root arbiter of trust within its domain; it is trusted with the secret key from which it derives all decryption keys for individuals within the organization. In an IBE system, an encrypted message may only be sent to a single entity within the domain of that specific TA. This paper considers a coalition of TA's desiring secure one-to-many communication, with each TA retaining the security of its secret keys. This secrecy requirement is natural in the setting of dynamic coalitions.

B. Known Extensions of IBE

A hierarchical identity-based encryption (HIBE) scheme [HL02] is an extension of identity-based encryption with identities organized in a tree with a single root master authority. An identity in non-hierarchical IBE is a unique single identifier, e.g. "Bob Smith serial number 123456789". An identity in a HIBE scheme is an ordered collection of identifiers which can mirror the organizational structure of the TA, e.g. ("US Army", "2-16 Infantry", "Bob"). Consider a two-level TA called the "US Army" with an analogous but simplified hierarchical structure. The first level of the hierarchy will identify a particular grouping of personnel, our examples use groupings at either the regiment or mission level, with the secret key for a grouping, e.g. ("US Army", "2-16 Infantry"), being held by its commanding officer. The second level will identify a specific individual or role within the grouping: our examples use individuals' names for this level. In a hierarchical IBE, as in regular IBE, encryption is accomplished by using the identity of a recipient in place of using the recipient's certified public key.

The hierarchical structure is meant to mirror the trust or authority structure of the organization. Any entity in a HIBE can use its own secret key to generate a secret key for any subordinate. If the head of the 2-16 infantry battalion has the secret key for identity ("US Army", "2-16 Infantry") then he is able to generate secret keys of the form ("US Army", "2-16 Infantry", "Name"). Messages may then be decrypted by any 'ancestor' capable of generating a secret key for the addressee, but a subordinate (or 'child') is not capable of generating a secret key for or decrypting messages addressed to any ancestor. This characteristic of HIBEs makes it absolutely necessary that the root authority of the hierarchy have the full faith and confidence of all members of the hierarchy. The root authority holds the master key for the HIBE scheme, making it capable of generating a secret key for any entity in the hierarchy and decrypting any message sent using the scheme.

One-to-many secure communication is a powerful cryptographic tool. In a MANETs setting, where transmission is much more expensive than computation, one-to-many communication allows for a single transmitted message to be read by any number of valid recipients within range. Multiple recipients may decrypt the same message through the use of one or more 'wildcards'. A wildcard is a special character, commonly denoted '*', that may be used in an address in lieu of specifying a particular aspect of an identity. Anyone with an identifier matching the non-wildcard portion is then able to read the message.

The method of employing wildcards into hierarchical identity-based encryption (WIBE) structures [ACD+06] allows any individual to send messages either to individual identities or to entire levels of a TA's hierarchy. To send a message to every member of a regiment in a HIBE, separate messages must be sent for individual in addresses as ("US Army", "2-16 Infantry", "Samuel Adams"). In a wildcard HIBE setting a single message could be addressed to ("US Army", "2-16 Infantry", *) which could then be read by each member of the regiment. A message addressed to the leadership of regiments could be sent to ("US Army", *) allowing for quick

[1]Graduate Center, City University of New York, USA.
[2]Queens College, City University of New York, USA.
[3]Royal Holloway, University of London, UK.

dissemination of command-level materials. Wildcards are also useful in the middle level of a hierarchy; a message can be addressed as ("US Army", *, "John Locke") to reach a "John Locke" regardless of his current regiment.

Other researchers have considered the question of developing IBE systems with multiple trusted authorities. For example, a scheme proposed by Paterson and Srinivasan [PS08] and another by Boklan *et al.* [BKPS08] introduced multi-TA IBE systems which allow trusted authorities to interact in order to derive a shared IBE system. The Paterson and Srinivasan [PS08] paper constructed an IBE scheme which supported multiple trust authorities in a way which makes it infeasible for an attacker to determine which trust authority's public parameters was used to form the ciphertext – i.e. the ciphertext preserves the anonymity of the trust authority. However, the Paterson and Srinivasan scheme does not allow trust authorities to form trust coalitions. The Boklan *et al.* scheme [BKPS08] allows trust authorities to cooperate to form trust coalitions, in the sense that within the coalition a private key issued by TA_i for an identity $\vec{ID}$ can be translated into a private key issued by TA_j for the same identity. However, in order to achieve this functionality, the scheme requires that the coalition TAs setup their master public parameters and master private keys simultaneously. Furthermore, every TA can deduce the master private key of every other TA. This is clearly a severe disadvantage for any setting where the TAs share anything less than complete trust in each others' intentions and security procedures. Neither scheme supports hierarchical identity-based encryption or one-to-many communication.

C. Our Contributions

This paper extends the concept of multi-TA IBE schemes to include hierarchical structures. For a set of uniquely named trust authorities ("US Army", "US Navy", "UK Forces", etc.) each with its own master secret used to create decryption keys for members of its hierarchy, the scheme put forward in this paper allows groups of these TA's to form coalitions without divulging any secret information. Communication within a coalition respects the original trust authority's hierarchy organizational structure, e.g. messages addressed to ("US Army", "1-75 Ranger Regiment", "William Perry") may be decrypted by the same individual when sent using the multi-TA coalition scheme or sent using the "US Army" HIBE.

We use the wildcard techniques from Abdalla *et al.* [ACD+06] to transition a coalition-based HIBE to a WIBE. Under this WIBE it is possible for anyone with a TA's public parameters to encrypt a message to any wildcard-based pattern of identities. Under any coalition configuration, all members of the "US Army" TA could be addressed as ("US Army", *, *) even if the sender was not a member of that TA. A sender possessing the public parameters of a coalition does not need to lie in the domain of trust of any TA in order to be able to send a message. A message may only be decrypted by members of the coalition under which it was encrypted, and only by those holding a decryption key matching the addressed wildcard pattern. A message may be addressed to a single specified TA or to all TA's in a coalition by placing wildcard ("$*$") in the TA-level identity. In our scheme, it is only possible to encrypt messages for one TA or all TA's within a coalition; however, a smaller subset of member TA's could be selected by forming another smaller coalition just with these TA's. Note that a message broadcast to all TA's in a coalition remains secure against non-members of the coalition.

Coalition formation is achieved through communication at the root level of trust authorities. Once a set of authorities agree to be members of a coalition, they publish information to make a coalition-specific public key and for each TA to adjust their keys for the coalition. This compatibility information may then be (publicly) broadcast to subordinates of the TA hierarchy enabling them to adapt their decryption keys for use within the coalition.

In this paper we offer an example instantiation based upon the Boneh-Boyen IBE [BB04] as extended to a WIBE by Abdalla *et al.* [ACD+06]. The system requires a bilinear map with certain properties. The Weil or Tate pairing would be a suitable example of such a map. When applied to supersingular elliptic curves, these pairings are also efficient enough for practical purposes. Included is a proof of security against selective-identity chosen-plaintext attacks in the standard model based upon the Bilinear Decisional Diffie-Hellman assumption (BDDH). In specific instantiations, including the one presented, coalitions may be dynamically reformed more efficiently than forming a coalition from scratch.

D. Usage Scenario

As an example of our techniques, we propose the following simplistic usage scenario. We suppose that individuals in a force can be identified using the hierarchical identifier ("force name", "mission name", "individual name"). For example, an individual could be identified as ("US force", "Mission 12", "John Smith"). Our scheme allows an architecture in which a central facility for each force generates a master key-pair. The master public key is publicly distributed and is intended for long-term use. We assume that all master public keys are known by all individuals in the scenario. The master private key is used by the central facility to derive keys for each hierarchical name ("force name", "mission name", "individual name"). These individual hierarchical encryption schemes can be used as independent WIBEs.

Our scheme allows coalitions to be formed between forces. For example, for a particular mission, the US and UK force may form a coalition. This allows messages to be sent to hierarchical identifiers of the form ("force name", "mission name", "individual name") where each identifier is either an individual name or a wildcard. In other words, to send a message to all individuals involved in the mission, one would merely have to send the message using the hierarchical name (*, "mission name", *) using the master public keys for the US and UK forces. This message could only be decrypted by an individual on the mission who has received the (public) broadcast update from their central facility which allows them to form the coalition key for the coalition containing the US forces and the UK forces.

Furthermore, these coalitions can be dynamic in nature. In the above scenario, if the German force was later included in

the mission, then broadcast messages could be sent to update coalition decryption keys to allow the US, UK, and German forces to decrypt messages. The German force members would not be able to read earlier messages, which were encrypted for the US and UK forces coalition only. If the US force retired from the mission, then the coalition keys could be updated so that only the UK and German forces could decrypt (future) messages.

One important restriction that we note about the above scenario, and seems to be common amongst all useful scenarios that we can envisage, is that the scheme requires some common agreement on naming. In the above scenario, the individual TA's are able to choose their own identifiers and the identifiers of the individuals at will; however, all TA's must agree on a common name for the mission. This allows users to send messages to patterns of the form (*,"mission name", *) that can be read by all authorised individuals.

II. Syntax

Throughout the paper we will use standard notation for algorithms and assignment. In particular, we will use $\leftarrow$ to denote assignment and $\xleftarrow{\$}$ to denote random assignment. Therefore, $y \leftarrow x$ denotes the assignment of the value x to the variable y and $y \xleftarrow{\$} S$ denotes the assignment of a uniform random element of the set S to the variable y. If $\mathcal{A}$ is a deterministic algorithm, then $y \leftarrow \mathcal{A}(x)$ denotes the assignment of the output of the algorithm $\mathcal{A}$ run on the input x to the variable y. Similarly, if $\mathcal{A}$ is a probabilistic algorithm, then $y \xleftarrow{\$} \mathcal{A}(x)$ denotes the assignment of the output of the algorithm $\mathcal{A}$ run on the input x to the variable y when $\mathcal{A}$ is computed using a fresh set of random coins.

A Trusted Authority is the root of a domain of trust with responsibilities over the namespace of its organization. In general we will refer to a Trusted Authority TA_i as a hierarchy of identities of the form $\vec{ID} = (ID_1, ID_2, \ldots, ID_k)$ with the same first identity ($ID_1 = TA_i$) and maximum depth of L such that $k \leq L$. Given a population of TA's $\mathcal{U} = \{TA_1, TA_2, \ldots, TA_n\}$ we define a coalition $\mathcal{C} = \{TA_a, TA_b, \ldots, TA_\ell\} \subseteq \mathcal{U}$. We also define a pattern to be a vector of identities and wildcards, i.e. $P = (P_1, \ldots, P_k)$ where $P_i \in \{0,1\}^* \cup \{*\}$ for $1 \leq i \leq k$. We say that an identity $\vec{ID} = (ID_1, \ldots, ID_k)$ matches a pattern $P = (P_1, \ldots, P_k)$, written $\vec{ID} \in_* P$, if $P_i \in \{ID_i, *\}$ for all $1 \leq i \leq k$. A multi-hierarchy WIBE consists of the following PPT algorithms/protocols:

- `CreateTA`: This algorithm creates the master public/private keys for a trust authority with a particular name ("TA_i"). This algorithm would be run by the central facility of a particular trust authority. The master public key is publicly distributed to everyone that would potentially send messages to the TA's hierarchy or form a coalition with the trust authority. The private key must be kept secret and is only used to create decryption keys for individuals in the hierarchy. Formally, the algorithm takes as input the proposed name for the trusted authority ("TA_i") and outputs a master key-pair (pk_i, sk_i), written $(pk_i, sk_i) \xleftarrow{\$} \texttt{CreateTA}(\text{"}TA_i\text{"})$.
- `SetupCoalitionBroadcast`: This algorithm allows a central facility to share the information necessary to initialize coalitions. Once a set of trust authorities agree to setup a coalition between them, each trust authority runs this algorithm to produce the information which allows the other trust authorities in the coalition to produce coalition keys for the members of its hierarchy. For some coalition $\mathcal{C} \subseteq \mathcal{U}$ containing TA_i, trust authority TA_i uses its secret key and the public keys of participating authorities to generate public parameters specific to each other authority. These parameters allow the other coalition members to include TA_i in a coalition. This is written as $W_i \xleftarrow{\$} \texttt{SetupCoalitionBroadcast}(TA_i, sk_i, \mathcal{C}, PK)$, where $PK = \{pk_j : TA_j \in \mathcal{C}\}$ is the set of master public keys in the coalition and $W_i = \{w_{i,j} : TA_j \in \mathcal{C} \setminus TA_i\}$ is the set of key update elements. Each $w_{i,j}$ is sent from TA_i to $TA_j \in \mathcal{C}$. These values are stored by TA_j for later use in the `SetupCoalitionKeys` algorithm.
- `SetupCoalitionKeys`: This algorithm completes the setup of the coalition $\mathcal{C} = \{TA_a, TA_b, \ldots, TA_\ell\} \subseteq \mathcal{U}$. After every member TA_i of the coalition has provided a message $w_{i,j}$ to TA_j, trust authority TA_j uses this algorithm to combine those messages to allow creation of coalition-specific secret keys. It outputs a message v_j to be broadcast to every member of TA_j's hierarchy, who then run the `ExtractCoalitionKey` algorithm to obtain their individual secret keys specific to coalition $\mathcal{C}$. Written, $v_j \xleftarrow{\$} \texttt{SetupCoalitionKeys}(TA_j, sk_j, \mathcal{C}, \hat{W}_j)$ where $\hat{W}_j = \{w_{i,j} : TA_i \in \mathcal{C} \setminus TA_j\}$ is the set of coalition parameters received by TA_j.

We now describe the algorithms required by the individual users.

- `Extract`: This algorithm is used by an individual in the domain of a trust authority to generate private keys for their subordinates (i.e. on the level below them in the hierarchical identity structure). This key-generating entity may be a member of any level of the hierarchy, except for the deepest level allowed (L), including the root authority. The keys generated are specific to the TA's HIBE, although they may later be adjusted for use in a coalition environment. For entity $\vec{ID} = (ID_1, ID_2, \ldots, ID_k)$ extracting a private key for subordinate $\vec{ID} \| ID' = (ID_1, ID_2, \ldots, ID_k, ID')$ the algorithm outputs $d_{\vec{ID}\|ID'} \xleftarrow{\$} \texttt{Extract}(\vec{ID}, d_{\vec{ID}}, ID')$.
- `ExtractCoalitionKey`: Users in a trust authority's hierarchy may use this algorithm to adapt their TA-specific HIBE private key for use within a coalition. To accomplish this their TA, TA_i, must provide an adjustment parameter v_i to be combined with the user's private key $d_{\vec{ID}}$. The adjustment parameter is specific to a particular coalition $\mathcal{C}$ containing TA_i, it is generated by the `SetupCoalitionKeys` algorithm. A user generates its coalition key as $c_{\vec{ID}} \leftarrow \texttt{ExtractCoalitionKey}(d_{\vec{ID}}, v_i)$. If the coalition key

is undefined, then we assume that $c_{\vec{ID}} = d_{\vec{ID}}$, i.e. the decryption key for the coalition which contains only the user's trust authority.

- `Encrypt`: This algorithm can be used by an individual to encrypt a message m to any individual whose identity matches a pattern P in the trust domain of a coalition $\mathcal{C}$. This is computed as $C \xleftarrow{\$} \texttt{Encrypt}(P, m, \mathcal{C}, PK)$ where $PK = \{pk_i : TA_i \in \mathcal{C}\}$.
- `Decrypt`: This algorithm can be used to decrypt a ciphertext C under a coalition key $c_{\vec{ID}}$ and outputs either a message m or the error symbol $\perp$. We write this operation as $\texttt{Decrypt}(ID, c_{\vec{ID}}, C)$. If no coalition is currently defined, then $c_{\vec{ID}} \leftarrow d_{\vec{ID}}$.

For correctness, we require that the decryption algorithm "undoes" the action of the encryption algorithm. Formally, we consider any collection of TA's $\mathcal{C} = \{TA_a, \ldots, TA_\ell\}$, with $(pk_i, sk_i) \xleftarrow{\$} \texttt{CreateTA}(TA_i)$, which form a coalition by computing

$$PK \leftarrow \{pk_i : TA_i \in \mathcal{C}\}$$

$$W_i \xleftarrow{\$} \texttt{SetupCoalitionBroadcast}(TA_i, sk_i, \mathcal{C}, PK) \text{ for all } TA_i \in \mathcal{C}$$

$$v_j \xleftarrow{\$} \texttt{SetupCoalitionKeys}(TA_j, sk_j, \mathcal{C}, \hat{W}_j) \text{ for all } TA_j \in \mathcal{C}\,.$$

Then for any identity $\vec{ID}$ and pattern P satisfying $\vec{ID} \in_* P$ and $ID_1 = TA_i \in \mathcal{C}$, we form the coalition key $c_{\vec{ID}}$ for $\vec{ID}$ by computing

$$d_{\vec{ID}} \xleftarrow{\$} \texttt{Extract}(TA_i, sk_i, \vec{ID})$$
$$c_{\vec{ID}} \xleftarrow{\$} \texttt{ExtractCoalitionKey}(d_{\vec{ID}}, v_i)\,.$$

In this situation, we demand that if $C \xleftarrow{\$} \texttt{Encrypt}(P, m, \mathcal{C}, PK)$ then $\texttt{Decrypt}(c_{\vec{ID}}, C) = m$.

III. Boneh-Boyen-Based Instantiation

We instantiate the notion of multi-TA WIBEs by extending the Boneh-Boyen WIBE [ACD+06]. Our scheme utilizes two (multiplicatively-written) cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ with prime order p for which there exists a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, i.e. we require that if g is a generator of $\mathbb{G}$ then $e(g, g) \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$. Furthermore, we assume that this map is efficiently computable. In practice, we can instantiate $\mathbb{G}$ as a prime-order subgroup of a supersingular elliptic curve group and $\mathbb{G}_T$ as a multiplicative subgroup of $\mathbb{F}_{p^r}$ for some value r, where the bilinear map is instantiated using the Weil or Tate pairing. For more information on this topic, the reader is referred to Galbraith, Paterson and Smart [GPS08].

Our scheme requires some general parameters which are used by all trust authorities. These include descriptions of the groups $(\mathbb{G}, \mathbb{G}_T)$, the prime p, and a description of the bilinear map e. We require that the general parameters include some randomly generated group elements. In particular, if the maximum hierarchy depth under each trust authority is $L-1$, then we require the group elements:

$$g_1, g_2 \xleftarrow{\$} \mathbb{G}^*$$
$$u_{i,j} \xleftarrow{\$} \mathbb{G}^* \text{ for } 1 \leq i \leq L \text{ and } j \in \{0, 1\}\,.$$

It is vitally important that the group elements are generated randomly and that no extra information about the group elements are revealed. In particular, the discrete logarithms of these group elements must be unknown. The group descriptions can be provided by a standardization body in a manner similar to the NIST curves [Nat99]. The group elements can be chosen using well-known techniques, for example by mapping a randomly chosen bitstring onto the curve. We let *param* denote these general parameters and assume that every algorithm takes these general parameters as an implicit input (even when not explicitly stated). Our scheme will regard identities as elements of $\mathbb{Z}_p$ and is described as follows:

- $\texttt{CreateTA}(TA_i)$: The TA generates $\alpha_i \xleftarrow{\$} \mathbb{Z}_p$ and computes master public key $pk_i \leftarrow g_1^{\alpha_i}$. The TA's private key is defined to be $sk_i \leftarrow g_2^{\alpha_i}$.
- $\texttt{Extract}(\vec{ID}, ID', d_{\vec{ID}})$: Recall that we regard a hierarchical identity as beginning with the trusted authority's identity; hence, in an identity tuple $\vec{ID}$, the first element ID_1 will always equal that identity's trusted authority TA_i. If the trusted authority $(TA_i) = \vec{ID}$ wishes to derive a key for a directly subordinate identity (TA_i, ID'), the trusted authority generates the private key $d_{(TA_i, ID')} = (h, a_1, a_2)$, where

  $$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p \qquad a_1 \leftarrow g_1^{r_1} \qquad a_2 \leftarrow g_1^{r_2}$$
  $$h \leftarrow g_2^{\alpha_i}(u_{1,0} \cdot u_{1,1}^{ID_1})^{r_1}(u_{2,0} \cdot u_{2,1}^{ID_2})^{r_2}\,.$$

  An entity with identity $\vec{ID} = (ID_1, ID_2, \ldots, ID_k)$ holding private key $d_{\vec{ID}} = (h, a_1, a_2, \ldots, a_k)$ may derive a private key for identity $(ID_1, ID_2, \ldots, ID_k, ID')$. This private key is generated as $d_{(ID_1, ID_2, \ldots, ID_k, ID')} = (h', a_1, a_2, \ldots, a_k, a_{k+1})$, where a_1 through a_k are retained and

  $$r \xleftarrow{\$} \mathbb{Z}_p \qquad a_{k+1} \leftarrow g_{k+1}^r$$
  $$h' \leftarrow h(u_{k+1,0} \cdot u_{k+1,1}^{ID'})^r\,.$$

- $\texttt{SetupCoalitionBroadcast}(TA_i, \mathcal{C})$: For $TA_j \in \mathcal{C}$, TA_i randomly generates $r_j \xleftarrow{\$} \mathbb{Z}_p$ and computes

  $$w_{i,j,0} \leftarrow g_2^{\alpha_i}(u_{0,0} \cdot u_{0,1}^{TA_j})^{r_j} \qquad w_{i,j,1} \leftarrow g_1^{r_j}$$

  where $TA_j \in \mathbb{Z}_p$ is the identity of TA_j. The algorithm sets $w_{i,j} = (w_{i,j,0}, w_{i,j,1})$ and outputs a list W_i of TA/message pairs $w_{i,j}\, \forall TA_j \in \mathcal{C} \setminus TA_i$.
- $\texttt{SetupCoalitionKeys}(TA_j, sk_j, \mathcal{C}, \hat{W}_j)$: After receiving all the messages $w_{i,j}$ from every $TA_i \in \mathcal{C}$, the algorithm outputs the TA_j-specific, coalition-specific adjustment parameter v_j as

  $$v_j \leftarrow \Big(\prod_{TA_i \in \mathcal{C} \setminus TA_j} w_{i,j,0}, \prod_{TA_i \in \mathcal{C} \setminus TA_j} w_{i,j,1}\Big)\,.$$

- $\texttt{ExtractCoalitionKey}(\mathcal{C}, v_j, d_{\vec{ID}})$: Parse v_j as $(v_{j,0}, v_{j,1})$. A user with private key $(h, a_0, a_1, a_2, \ldots, a_\ell)$ can form a coalition key $(h', a_0', a_1, a_2, \ldots, a_\ell)$ where

$$\begin{aligned} h' &\leftarrow h \cdot v_{j,0} \\ &= h \prod_{TA_i \in \mathcal{C} \setminus TA_j} w_{i,0} \\ &= g_2^{\sum \alpha_i} (u_{0,0} \cdot u_{0,1}^{TA_j})^{\sum r_i} \\ a_0' &\leftarrow a_0 \cdot v_{j,1} \\ &= a_0 \prod_{TA_i \in \mathcal{C} \setminus TA_j} w_{i,1} = g_1^{r + \sum r_i}. \end{aligned}$$

- $\texttt{Encrypt}(P, m, \mathcal{C}, PK)$: Let ℓ be the depth of the pattern P and $W(P)$ be the set of levels which have wildcard characters. The sender chooses $t \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext $C = (C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$ where

$$\begin{aligned} C_1 &\leftarrow g_1^t \\ C_{2,i} &\leftarrow \begin{cases} (u_{i,0} \cdot u_{i,1}^{P_i})^t & \text{if } i \notin W(P) \\ (u_{i,0}^t, u_{i,1}^t) & \text{if } i \in W(P) \end{cases} \\ C_3 &\leftarrow m \cdot e(\prod_{TA_j \in \mathcal{C}} pk_j, g_2)^t \end{aligned}$$

  The ciphertext produced here is coalition-specific as it depends on the public keys of all the trust authorities in the coalition. A wildcard may be used at the TA identifier level (i.e. $P_1 = *$) to address all the trust authorities in the coalition.
- $\texttt{Decrypt}(\vec{ID}, c_{\vec{ID}}, C)$: Parse $\vec{ID}$ as $(ID_1, \ldots, ID_\ell)$, c_{ID} as $(h, a_1, \ldots, a_\ell)$, and C as $(C_1, C_{2,1}, \ldots, C_{2,\ell}, C_3)$. For each $i \in W(P)$, parse $C_{2,i}$ as $(\tilde{u}_{i,0}, \tilde{u}_{i,1})$. We recover a complete HIBE ciphertext by setting $C_{2,i}' \leftarrow C_{2,i}$ if $i \notin W(P)$, and $C_{2,i}' \leftarrow \tilde{u}_{i,0} \cdot \tilde{u}_{i,1}^{ID_i}$ if $i \in W(P)$. Recover

$$m' \leftarrow C_3 \frac{\prod_{i=1}^{\ell} e(a_i, C_{2,i}')}{e(C_1, h)}$$

  and return m'.

A multi-trust-authority HIBE scheme may be designed analogously except that we remove the possibility of the pattern containing wildcards. For both schemes, we note that we can quickly reconfigure a coalition by re-using existing values of $w_{i,j}$ between trust authorities that have previously exchanged coalition messages.

IV. Security Results

A. Security Model

We provide a formal mathematical model for the selective-identity IND-CPA security of a multi-TA WIBE. This is a weak notion of security, but serves as a basis for assessing the security of the Boneh-Boyen scheme and as a basis for developing more complex security models. A selective-identity IND-CPA security model for a multi-TA HIBE can be defined in an analogous way.

The aim of the security model is to capture the notion that an attacker (who wishes to attack messages sent to a particular individual in a known coalition) cannot obtain any information about messages simply by observing the ciphertexts as they are transmitted. As we said, this is a very simple notion of security; it does not prevent an attacker from determining information about a message if it can (a) interact with the system by (for example) obtaining some information about the messages underlying other ciphertexts, or (b) dynamically decide which ciphertext to attack during the course of the attack. However, this security notion is sufficient to demonstrate that the Boneh-Boyen approach is useful and standard techniques to immunize the scheme against the more sophisticated attacks given above are available in the full version of the paper.

The security model is presented as a game played between a probabilistic polynomial-time attacker $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and a hypothetical challenger. The sID-IND-CPA security game runs as follows:

1) The attacker runs $\mathcal{A}_0(1^k)$. The attacker outputs a challenge coalition $\mathcal{C}^*$ and a challenge patter $P^* = (P_1^*, \ldots, P_k^*)$, where $P_1^* \in \mathcal{C}^*$ or $P_1^* = *$, along with some state information $state$.
2) The challenger generates any general parameters $param$ and public/private keys for each trust authority in the challenge coalition $(pk_i, sk_i) \xleftarrow{\$} \texttt{CreateTA}(\text{“}TA_i\text{”})$ (for $TA_i \in \mathcal{C}^*$).
3) The attacker runs $\mathcal{A}_1$ on the input $(param, PK, state)$ where $PK = \{pk_i : TA_i \in \mathcal{C}^*\}$. $\mathcal{A}_1$ may query the following oracles during its execution:
   - $\texttt{CreateTA}(TA)$: The oracle computes $(pk_i, sk_i) \xleftarrow{\$} \texttt{Setup}(1^k, TA)$ for the TA identity TA_i and returns pk_i. This oracle can only be queried once for each identity TA_i.
   - $\texttt{SetupCoalitionBroadcast}(TA_i, \mathcal{C})$: This oracle runs the $\texttt{SetupCoalitionBroadcast}$ algorithm for coalition $\mathcal{C}$ containing TA_i and returns messages $w_{i,j}$ for all $TA_j \in \mathcal{C} \setminus TA_i$.
   - $\texttt{SetupCoalitionKeys}(TA_j, \mathcal{C}, \hat{W}_j)$: The oracle runs the $\texttt{SetupCoalitionKeys}$ algorithm assuming that message $w_{i,j}$ was sent by TA_i and returns the resulting message v_j. Note that this does not imply that all the TAs believe that they're in the same coalition or that $w_{i,j}$ has been return by the $\texttt{SetupCoalitionBroadcast}$ oracle.
   - $\texttt{Corrupt}(\vec{ID})$: The oracle returns $d_{\vec{ID}}$ for the identity $\vec{ID}$. Note that if $\vec{ID} = TA_i$ then this method returns TA_i's secret key sk_i.

   The attacker $\mathcal{A}_1$ terminates with the output of two equal-length messages (m_0, m_1) and some state information $state$.
4) The challenger randomly chooses a bit $b \xleftarrow{\$} \{0, 1\}$ and computes the ciphertext $C^* \xleftarrow{\$} \texttt{Encrypt}(P^*, m_b, \mathcal{C}^*, PK)$ where $PK = \{pk_i : TA_i \in \mathcal{C}^*\}$.
5) The attacker runs $\mathcal{A}_2$ in the input $(C^*, state)$. The attacker $\mathcal{A}_2$ may query the same oracle in the previous phase of the security game. The attacker terminates with the output of a bit b'.

The attacker wins the game if $b = b'$ and the attacker did not make a $\texttt{Corrupt}$ query for any $\vec{ID}$ for which there exists

$\vec{ID}'$ such that $\vec{ID}\|\vec{ID}' \in_* P^*$. An attacker which did not make such an oracle query is defined to have an advantage:

$$Adv_{\mathcal{A}}^{\mathtt{IND}}(k) = |\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]|.$$

B. The Security of the Boneh-Boyen Multi-TA WIBE

The intuition behind the multi-TA Boneh-Boyen scheme is that any coalition of trust authorities can be viewed as an extended hierarchy with a "ghost" trust authority at the top level. Each actual trust authority is represented as a first-level identity under this ghost TA and, through communication with the other trust authorities in the coalition, is able to determine a private key for their first-level identity under the ghost TA. More specifically, if we consider a coalition $\mathcal{C} = \{TA_1, \ldots, TA_n\}$ in which each TA has a private key $sk_i = g_2^{\alpha_i}$, then the ghost TA will have a private key $g_2^{\sum \alpha_i}$. Upon forming the coalition, the trust authority TA_j receives the messages

$$w_{i,j,0} \leftarrow g_2^{\alpha_i}(u_{0,0} \cdot u_{0,1}^{TA_j})^{r_i} \qquad w_{i,j,1} \leftarrow g_1^{r_i}$$

from each $TA_i \in \mathcal{C} \setminus \{TA_j\}$. This allows TA_j to form the private key

$$h \leftarrow g_2^{\sum_i \alpha_i}(u_{0,0} \cdot u_{0,1}^{TA_j})^{\sum_{i \neq j} r_i} \qquad a_1 \leftarrow g_1^{\sum_{i \neq j} r_i}$$

which is precisely the key that would be obtained if the ghost TA were to distribute a private key to the identity TA_j. The security of the multi-TA scheme then essentially follows from the security of the single-TA WIBE scheme, although care must be taken to show that the broadcast messages $w_{i,j}$ and v_i do not leak information about the private keys to the attacker.

The Boneh-Boyen multi-TA WIBE scheme we have presented is secure in the sID-IND-CPA security model under the Bilinear Decision Diffie-Hellman (BDDH) problem. This can be defined as:

Definition 1: Let p be a prime of length k, let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of order p, let g a generator for $\mathbb{G}$, and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. Let D_k be the distribution $\vec{x} \leftarrow (g, g^a, g^b, g^c, e(g,g)^{abc})$ where $a, b, c \xleftarrow{\$} \mathbb{Z}_p$. Let R_k be the distribution $\vec{x} \leftarrow (g, g^a, g^b, g^c, Z)$ where $a, b, c \xleftarrow{\$} \mathbb{Z}_p$ and $Z \xleftarrow{\$} \mathbb{G}_T$. An algorithm $\mathcal{A}$ has advantage $Adv_{\mathcal{A}}^{\mathtt{BDDH}}(k)$ against the Bilinear Decision Diffie-Hellman problem if

$$Adv_{\mathcal{A}}^{\mathtt{BDDH}}(k) = |\Pr[\mathcal{A}(\vec{x}) = 1 \,|\, \vec{x} \xleftarrow{\$} D_k] - \Pr[\mathcal{A}(\vec{x}) = 1 \,|\, \vec{x} \xleftarrow{\$} R_k]|.$$

We shall assume that it is infeasible for any probabilistic polynomial-time attacker to solve the BDDH problem on the groups $(\mathbb{G}, \mathbb{G}_T)$.

We prove the following theorems in the full version of the paper:

Theorem 2: Suppose that there exists an attacker $\mathcal{A}$ against the selective-identity multiple TA sID-IND-CPA Boneh-Boyen WIBE that runs in time t and has advantage $Adv_{\mathcal{A}}^{\mathtt{WIBE}}(k)$, then there exists an attacker $\mathcal{B}$ against the selective-identity multiple TA sID-IND-CPA security of the Boneh-Boyen HIBE that runs in time t' and has advantage $Adv_{\mathcal{B}}^{\mathtt{HIBE}}(k)$ where $t \approx t'$ and $Adv_{\mathcal{A}}^{\mathtt{WIBE}}(k) = Adv_{\mathcal{B}}^{\mathtt{HIBE}}(k)$.

Theorem 3: Suppose that there exists an attacker $\mathcal{A}$ against the selective-identity multiple TA sID-IND-CPA security of the Boneh-Boyen HIBE that runs in time t, makes at most q_K queries to the `Corrupt` oracle, and has advantage $Adv_{\mathcal{A}}^{\mathtt{HIBE}}(k)$, then there exists an algorithm $\mathcal{B}$ that solves the DBDH problem that runs in time $t' = O(t)$ and has advantage $Adv_{\mathcal{B}}^{\mathtt{BDDH}}(k) \geq Adv_{\mathcal{A}}^{\mathtt{HIBE}}(k)/2 - q_K/2p$.

We therefore conclude the following:

Corollary 4: If the BDDH problem is difficult, then the Boneh-Boyen multiple TA WIBE scheme is secure in the sID-IND-CPA security model.

V. Acknowledgements

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[ACD+06] M. Abdalla, D. Catalano, A. W. Dent, J. Malone-Lee, G. Neven, and N. P Smart. Identity-based encryption gone wild. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, pages 300–311. Springer-Verlag, 2006.

[BB04] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT '04*, pages 223–238. Springer-Verlag, 2004.

[BKPS08] K. D. Boklan, Z. Klagsbrun, K. G. Paterson, and S. Srinivasan. Flexible and secure communications in an identity-based coalition environment. In *Proc. IEEE Military Communications Conference – MILCOM 2008*, 2008.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156:3113–3121, 2008.

[HL02] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT '02*, pages 466–481. Springer-Verlag, 2002.

[Nat99] National Institute of Standards and Technology. *Recommended Elliptic Curves for Federal Government Use*, 1999.

[PS08] K. G. Paterson and S. Srinivasan. Security and anonymity of identity-based encryption with multiple trusted authorities. In S. D. Galbraith and K. G. Paterson, editors, *Pairing '08: Proceedings of the 2nd international conference on Pairing-Based Cryptography*, pages 354–375. Springer-Verlag, 2008.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – CRYPTO '84*, pages 47–53. Springer-Verlag, 1984.