Key Indistinguishability vs. Strong Key Indistinguishability for Hierarchical Key Assignment Schemes

Arcangelo Castiglione, Alfredo De Santis and Barbara Masucci

Abstract

A *hierarchical key assignment scheme* is a method to assign some private information and encryption keys to a set of classes in a partially ordered hierarchy, in such a way that the private information of a higher class can be used to derive the keys of all classes lower down in the hierarchy.

In this paper we analyze the security of hierarchical key assignment schemes according to different notions: security with respect to *key indistinguishability* and against *key recovery*, as well as the two recently proposed notions of security with respect to *strong key indistinguishability* and against *strong key recovery*. We first explore the relations between all security notions and, in particular, we prove that security with respect to strong key indistinguishability is *not stronger* than the one with respect to key indistinguishability. Afterwards, we propose a general construction yielding a hierarchical key assignment scheme offering security against strong key recovery, given any hierarchical key assignment scheme which guarantees security against key recovery.

Index Terms

Access control, key assignment, provable security, key indistinguishability, strong key indistinguishability, key recovery, strong key recovery.

I. Introduction

THE *access control management* ensures that only authorized users are given access to certain resources. In particular, with respect to their respective powers and responsibilities, users are typically organized into *hierarchies*, composed by several disjoint classes (*security classes*). Hierarchical structures are widely employed in many different application areas, including database management systems, computer networks, operating systems, military and government communications.

The use of cryptographic techniques to address the problem of key management in hierarchical structures has been first considered by Akl and Taylor [1], who proposed a *hierarchical key assignment scheme* where each class

The authors are with the Dipartimento di Informatica, Università di Salerno, Fisciano, Salerno, 84084, Italy (e-mails: {arcastiglione, ads, bmasucci}@unisa.it.)

is assigned a key that can be used, along with some public information generated by a trusted authority, to compute the key assigned to any class lower down in the hierarchy. Subsequently, many researchers have proposed schemes offering different trade-offs in terms of the amount of public and private information and the complexity of key derivation (e.g., [2]–[18]). Many other proposals either support more general access control policies [19]–[22] or satisfy additional time-dependent constraints [23]–[32]. Despite the large number of proposed schemes, many of them lack a formal security proof and have been shown to be insecure against *collusive attacks* [27], [33]–[36], whereby two or more classes collude to compute a key to which they are not entitled.

According to the *security reduction paradigm* introduced by Goldwasser and Micali [37], a scheme is *provably-secure* under a complexity assumption if the existence of an adversary A breaking the scheme is equivalent to the existence of an adversary B breaking the computational assumption [37]. Atallah et al. [14] first addressed the problem of formalizing security requirements for hierarchical key assignment schemes and proposed two different notions: security against *key recovery* and with respect to *key indistinguishability*. Informally speaking, the former captures the notion that an adversary should not be able to compute a key to which it should not have access, while in the latter, the adversary should not even be able to distinguish between the real key and a random string of the same length. In particular, the model considered in [14] allows an adversary attacking a certain class in the hierarchy to gain access to the private information assigned to all users not allowed to access such class, as well as all the public information.

Atallah et al. [14] also proposed two provably-secure constructions for hierarchical key assignment schemes: the first one is based on pseudorandom functions and satisfies security against key recovery, whereas, the second one requires the additional use of a symmetric encryption scheme and guarantees security with respect to key indistinguishability. Different constructions satisfying the above defined notions of security have been proposed in [12], [15]–[18], [27], [31], [32], [38]. In particular, De Santis et al. [12], [17] proposed two different constructions satisfying security with respect to key indistinguishability: the first one, which is based on symmetric encryption schemes, is simpler than the one proposed in [14], requires a single computational assumption, and offers more efficient procedures for key derivation and key updates; the second one, which is based on a public-key broadcast encryption scheme, allows to obtain a hierarchical key assignment scheme offering constant private information and public information linear in the number of classes. D'Arco et al. [15], [16] analyzed the Akl-Taylor scheme according to the definitions proposed in [14] and showed how to choose the public parameters in order to get instances of the scheme which are secure against key recovery under the RSA assumption. Moreover, they showed how to turn the Akl-Taylor scheme in a construction offering security with respect to key indistinguishability; however such a scheme, is less efficient than the constructions proposed in [12], [14], [17]. Freire et al. [18] proposed a construction based on pseudorandom generators, satisfying security with respect to key indistinguishability. Finally, Ateniese et al. [27], [32] extended the model proposed in [14] to schemes satisfying additional time-dependent constraints and proposed two different constructions offering security with respect to key indistinguishability. Other constructions for time-dependent schemes, offering different trade-offs in terms of amount of public and private information and complexity of key derivation, were shown in [30], [31], [38], [39]. Recently, a more general scenario has been considered for hierarchical key assignment schemes [40]. In such a scenario, the access control is not only hierarchical, but also *shared* between different classes. In particular, the authors of [40] proposed a construction for hierarchical and shared key assignment schemes that is secure with respect to key indistinguishability and relies on both *symmetric encryption* and *perfect secret sharing*.

Freire et al. [41] proposed new security definitions for hierarchical key assignment schemes. Such definitions, called security against *strong key recovery* and security with respect to *strong key indistinguishability*, provide the adversary with additional compromise capability, thus representing a strengthening of the model provided in [14]. As stated by Freire et al., such a new model is able to characterize a variety of scenarios which may arise in real-world situations, since it allows the protection of the key assigned to a certain class u, even when the keys held by classes which are predecessors of u in the hierarchy have been leaked, due to their use, loss or theft. More precisely, Freire et al. considered an adversary which, given a certain class, is allowed to gain the private information assigned to all users not allowed to access such class, as well as all the public information and *encryption keys assigned to all the other classes which are predecessors of the target class in the hierarchy*. Freire et al. also proposed two hierarchical key assignment schemes which are secure in the sense of strong key indistinguishability. The first construction is based on *pseudorandom functions*, whereas, the second one is based on *forward-secure pseudorandom generators*. Finally, they showed that the notions of security against key recovery and against strong key recovery *are separated*, i.e., there exist schemes that are secure against key recovery but which are not secure against strong key recovery. On the other hand, they did not clarify the relations between the notions of security with respect to key indistinguishability and with respect to strong key indistinguishability.

In this work, we explore the relations between all security notions for hierarchical key assignment schemes, by clarifying implications and separations occurring between such notions. In particular, we show that security with respect to strong key indistinguishability is *not stronger* than the one with respect to key indistinguishability, thus establishing the equivalence between such two security notions. A similar result has been recently shown in the *unconditionally secure setting* [42]. Furthermore, we also show how to construct a hierarchical key assignment scheme which is secure against strong key recovery, starting from any scheme which guarantees security against key recovery.

The paper is organized as follows: in Section II we review the definition of hierarchical key assignment schemes; in Section III we describe all security definitions for hierarchical key assignment schemes; in Section IV we analyze the relations among these definitions and in particular we show that security with respect to strong key indistinguishability is *not stronger* than the one with respect to key indistinguishability; finally in Section V, we show how to construct a hierarchical key assignment scheme secure against strong key recovery, starting from any hierarchical key assignment scheme which is secure against key recovery.

II. Hierarchical Key Assignment Schemes

Consider a set of users divided into a number of disjoint classes, called *security classes*. A security class can represent a person, a department or a user group in an organization. A binary relation $\preceq$ that partially orders the set of classes V is defined in accordance with authority, position or power of each class in V. The poset $(V, \preceq)$ is called a *partially ordered hierarchy*. For any two classes u and v, the notation $u \preceq v$ is used to indicate that the users in v can access u's data. Clearly, since v can access its own data, it holds that $v \preceq v$, for any $v \in V$. We denote the accessible set of a class v by A_v, which corresponds to the set $\{u \in V : u \preceq v\}$, for any $v \in V$. The partially ordered hierarchy $(V, \preceq)$ can be represented by the directed graph $G^* = (V, E^*)$, where each class corresponds to a vertex in the graph and there is an edge from class v to class u if and only if $u \preceq v$. We denote by $G = (V, E)$ the *minimal representation* of the graph G^*, namely, the directed acyclic graph corresponding to the *transitive and reflexive reduction* of the graph $G^* = (V, E^*)$. The graph G has the same transitive and reflexive closure of G^*, i.e., there is a path (of length greater than or equal to zero) from v to u in G if and only if there is the edge (v, u) in E^*. Aho et al. [43] showed that every directed graph has a transitive reduction, which can be computed in polynomial time and is unique for directed acyclic graphs. In the following, we denote by Γ a family of graphs corresponding to partially ordered hierarchies. For example, Γ could be the family of the rooted trees [7], the family of the d-dimensional hierarchies [13], etc..

A hierarchical key assignment scheme for a family Γ of graphs, corresponding to partially ordered hierarchies, is defined as follows in [12], [15]–[17], [27], [30]–[32].

Definition 2.1: A *hierarchical key assignment scheme* for Γ is a pair (Gen, Der) of algorithms satisfying the following conditions:

1) The *information generation algorithm Gen* is probabilistic polynomial-time. It takes as inputs the security parameter 1^τ and a graph $G = (V, E)$ in Γ, and produces as outputs

   a) a private information s_u, for any class $u \in V$;

   b) a key k_u, for any class $u \in V$;

   c) a public information pub.

   We denote by (s, k, pub) the output of the algorithm Gen on inputs 1^τ and G, where s and k respectively denote the sequences of private information and keys.

2) The *key derivation algorithm Der* is deterministic polynomial-time. It takes as inputs the security parameter 1^τ, a graph $G = (V, E)$ in Γ, two classes u, v in V, the private information s_u assigned to class u and the public information pub, and produces as output the key k_v assigned to class v if $v \in A_u$, or a special rejection symbol $\perp$ otherwise.

   We require that for each class $u \in V$, each class $v \in A_u$, each private information s_u, each key k_v, each public information pub which can be computed by Gen on inputs 1^τ and G, it holds that

$$Der(1^\tau, G, u, v, s_u, pub) = k_v.$$

The efficiency of a hierarchical key assignment scheme is evaluated according to different parameters: storage requirements, which correspond to the amount of secret data that needs to be distributed and stored by the users and the amount of data that needs to be made public; the complexity of both key derivation and key update procedures (it is desirable that updates to the access hierarchy require only local changes to the public information and do not need any private information to be re-distributed); the computational assumption on which the security of the scheme relies (it is desirable to employ standard assumptions).

III. Notions of Security

A hierarchical key assignment scheme must be resistant to *collusive attacks*. More precisely, for each class $u \in V$, the key k_u should be protected against a coalition of all users in the set $F_u = \{v \in V : u \notin A_v\}$, corresponding to the ones which are not allowed to compute the key k_u.

Atallah et al. [14] first introduced two different security goals for hierarchical key assignment schemes: security with respect to *key-indistinguishability* and security against *key recovery*. The former formalizes the requirement that the adversary is not able to learn any information (even a single bit) about a key k_u which it should not have access to, i.e., it is not able to distinguish it from a random string having the same length. On the other hand, the latter corresponds to the weaker requirement that an adversary is not able to compute a key k_u which it should not have access to. The notion of key indistinguishability offers security guarantees that cannot be achieved by schemes whose security relies only upon key recovery. These stronger security guarantees could be necessary. For example, as pointed out in [17], it is straightforward that the key indistinguishability notion is needed when the data associated to a class are protected by means of a symmetric encryption scheme, whose implementation details make the confidentiality of the ciphertext (or of part of it) depending on the secrecy of only a portion of the encryption key.

Recently, Freire et al. [41] proposed a new security definition for hierarchical key assignment schemes. Such a definition, called security with respect to *strong key-indistinguishability*, formalizes the requirement that the adversary is not able to learn any information about a key k_u which it should not have access to, *even if it has the additional capability of gaining access to the encryption keys associated to all other classes which are predecessors of the target class in the hierarchy*. Notice that these encryption keys might leak through usage and their compromise could not directly lead to a compromise of the private information s_u or the encryption key k_u of the target class u. Freire et al. also introduced the definition of security against *strong key recovery*. Such a definition formalizes the requirement that the adversary is not able to compute a key k_u which it should not have access to, even if it has the additional capability of gaining access to encryption keys assigned to all the other classes which are predecessors of the target class in the hierarchy.

In the following, we consider a *static* adversary which, given a class u, is allowed to gain the private information assigned to all users not allowed to access such class, as well as all the relative public information. For the case of *strong key indistinguishability* and *strong key recovery*, such an adversary is also able to access keys assigned to all other classes which are predecessors of the target class in the hierarchy. A different kind of adversary, the *adaptive* one, could be also considered. In detail, such an adversary is first allowed to access all public information as well as all private information of a number of classes of its choice; afterwards, it chooses the class u it wants to attack. In [27], [32] it has been proved that security with respect to adaptive adversaries is (*polynomially*) equivalent to the one against static ones. In particular, the scenario considered in [27], [32], [44] is more general, since the lifetime of each key is limited to a given period of time. In such a setting, each class is assigned to a different key for each different period of time. These schemes are called *Time-Bound Hierarchical Key Assignment Schemes*. However, the equivalence between adaptive and static adversaries shown in [27], [32] also applies to hierarchical key assignment schemes, since they can be seen as time-bound hierarchical key assignment schemes with a single period of time. Therefore, in this paper we will only consider static adversaries.

We use the standard notation to describe probabilistic algorithm and experiments following [45]. If $A(\cdot, \cdot, \ldots)$ is any probabilistic algorithm then $a \leftarrow A(x, y, \ldots)$ denotes the experiment of running A on inputs $x, y, \ldots$ and letting a be the outcome, the probability being over the coin tosses of A. Similarly, if X is a set then $x \leftarrow X$ denotes the experiment of selecting an element uniformly from X and assigning x this value. If w is neither an algorithm nor a set, then $x \leftarrow w$ is a simple assignment statement. A function $\epsilon : N \rightarrow R$ is *negligible* if for every constant $c > 0$ there exists an integer n_c such that $\epsilon(n) < n^{-c}$ for all $n \geq n_c$.

A. Security w.r.t. Key Indistinguishability

Consider a *static adversary* $\mathtt{STAT}_u$ that wants to attack a class $u \in V$ and which is able to corrupt *all* users in F_u. We define an algorithm $Corrupt_u$, which on input the private information s generated by the algorithm Gen, extracts the secret values s_v associated to all classes $v \in F_u$. We denote by $corr_u$ the sequence output by $Corrupt_u(s)$. Two experiments are considered. In the first one, the adversary is given the key k_u, whereas, in the second one, it is given a random string ρ having the same length as k_u. It is the adversary's job to determine whether the received challenge corresponds to k_u or to a random string. We require that the adversary will succeed with probability only negligibly different from $1/2$.

Definition 3.1: [IND−ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E)$ be a graph in Γ, let (Gen, Der) be a hierarchical key assignment scheme for Γ and let $\mathtt{STAT}_u$ be a static adversary which attacks a class u. Consider the following two experiments:

```
Experiment $\mathbf{Exp}^{\mathtt{IND-1}}_{\mathtt{STAT}_u}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $d \leftarrow \mathtt{STAT}_u(1^\tau, G, pub, corr_u, k_u)$
    return $d$

Experiment $\mathbf{Exp}^{\mathtt{IND-0}}_{\mathtt{STAT}_u}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $\rho \leftarrow \{0, 1\}^{length(k_u)}$
    $d \leftarrow \mathtt{STAT}_u(1^\tau, G, pub, corr_u, \rho)$
    return $d$
```

The advantage of $\texttt{STAT}_u$ is defined as

$$\begin{aligned}\mathbf{Adv}^{\texttt{IND}}_{\texttt{STAT}_u}(1^\tau, G) = |&Pr[\mathbf{Exp}^{\texttt{IND-1}}_{\texttt{STAT}_u}(1^\tau, G) = 1] \\ &- Pr[\mathbf{Exp}^{\texttt{IND-0}}_{\texttt{STAT}_u}(1^\tau, G) = 1]|.\end{aligned}$$

The scheme is said to be *secure in the sense of* IND-ST if, for each graph $G = (V, E)$ in Γ and each $u \in V$, the function $\mathbf{Adv}^{\texttt{IND}}_{\texttt{STAT}_u}(1^\tau, G)$ is negligible, for each static adversary $\texttt{STAT}_u$ whose time complexity is polynomial in τ.

B. Security against Key Recovery

Now consider the case where there is a static adversary $\texttt{STAT}_u$ which wants to *compute* the key assigned to a class $u \in V$. As done before, we denote by $corr_u$ the sequence output by the algorithm $Corrupt_u$, on input the private information s generated by the algorithm Gen. The adversary, on input all public information generated by the algorithm Gen, as well as the private information $corr_u$, outputs a string k'_u and succeeds whether $k'_u = k_u$. We require that the adversary will succeed with probability only negligibly different from $1/2^{length(k_u)}$.

Definition 3.2: [REC-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E)$ be a graph in Γ, let (Gen, Der) be a hierarchical key assignment scheme for Γ and let $\texttt{STAT}_u$ be a static adversary which attacks a class u. Consider the following experiment:

```
Experiment $\mathbf{Exp}^{\texttt{REC}}_{\texttt{STAT}_u}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $k'_u \leftarrow \texttt{STAT}_u(1^\tau, G, pub, corr_u)$
    return $k'_u$
```

The advantage of $\texttt{STAT}_u$ is defined as

$$\mathbf{Adv}^{\texttt{REC}}_{\texttt{STAT}_u}(1^\tau, G) = Pr[k'_u = k_u].$$

The scheme is said to be *secure in the sense of* REC-ST if, for each graph $G = (V, E)$ in Γ and each class $u \in V$, the function $\mathbf{Adv}^{\texttt{REC}}_{\texttt{STAT}_u}(1^\tau, G)$ is negligible, for each static adversary $\texttt{STAT}_u$ whose time complexity is polynomial in τ.

C. Security w.r.t. Strong Key Indistinguishability

Consider a *static adversary* $\texttt{STAT}_u$ that wants to attack a class $u \in V$. Such adversary is able to corrupt *all* users in F_u and to gain access to the keys associated to all classes in the set $P_u = \{v \in V \setminus \{u\} : u \in A_v\}$ of the predecessors of class u. As done before, we denote by $corr_u$ the sequence output by the algorithm $Corrupt_u$, on input the private information s generated by the algorithm Gen. Moreover, we define an algorithm $Keys_u$, which on input the encryption keys k generated by the algorithm Gen, extracts keys k_v associated to all classes $v \in P_u$.

We denote by $keys_u$ the sequence output by $Keys_u(k)$. Two experiments are considered. In the first one, the adversary is given the key k_u, whereas, in the second one, it is given a random string ρ having the same length as k_u. It is the adversary's job to determine whether the received challenge corresponds to k_u or to a random string. We require that the adversary will succeed with probability only negligibly different from $1/2$.

Definition 3.3: [STRONG-IND-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E)$ be a graph in Γ, let (Gen, Der) be a hierarchical key assignment scheme for Γ and let $\texttt{STAT}_u$ be a static adversary which attacks a class u. Consider the following two experiments:

```
Experiment $\mathbf{Exp}_{\texttt{STAT}_u}^{\texttt{STRONG-IND-1}}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $keys_u \leftarrow Keys_u(k)$
    $d \leftarrow \texttt{STAT}_u(1^\tau, G, pub, corr_u, keys_u, k_u)$
    return $d$
```

```
Experiment $\mathbf{Exp}_{\texttt{STAT}_u}^{\texttt{STRONG-IND-0}}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $keys_u \leftarrow Keys_u(k)$
    $\rho \leftarrow \{0,1\}^{length(k_u)}$
    $d \leftarrow \texttt{STAT}_u(1^\tau, G, pub, corr_u, keys_u, \rho)$
    return $d$
```

The advantage of $\texttt{STAT}_u$ is defined as

$$\begin{aligned}\mathbf{Adv}_{\texttt{STAT}_u}^{\texttt{STRONG-IND}}(1^\tau, G) = |&Pr[\mathbf{Exp}_{\texttt{STAT}_u}^{\texttt{STRONG-IND-1}}(1^\tau, G) = 1] \\ &- Pr[\mathbf{Exp}_{\texttt{STAT}_u}^{\texttt{STRONG-IND-0}}(1^\tau, G) = 1]|.\end{aligned}$$

The scheme is said to be *secure in the sense of* STRONG-IND-ST if, for each graph $G = (V, E)$ in Γ and each $u \in V$, the function $\mathbf{Adv}_{\texttt{STAT}_u}^{\texttt{STRONG-IND}}(1^\tau, G)$ is negligible, for each static adversary $\texttt{STAT}_u$ whose time complexity is polynomial in τ.

D. Security against Strong Key Recovery

Finally, consider the case where there is a static adversary $\texttt{STAT}_u$ that wants to *compute* the key assigned to a class $u \in V$. Such adversary is able to corrupt *all* users in F_u and gain access to the keys associated to all classes in the set P_u of the predecessors of u. As done before, we denote by $corr_u$ the sequence output by the algorithm $Corrupt_u$, on input the private information s generated by the algorithm Gen. Moreover, we denote by $keys_u$ the sequence output by $Keys_u(k)$. The adversary, on input all public information generated by the algorithm Gen, as well as the private information $corr_u$ and the sequence $keys_u$, outputs a string k'_u and succeeds if $k'_u = k_u$. We require that the adversary will succeed with probability only negligibly different from $1/2^{length(k_u)}$.

Definition 3.4: [STRONG-REC-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E)$ be a graph in Γ, let (Gen, Der) be a hierarchical key assignment scheme for Γ and let $\mathtt{STAT}_u$ be a static adversary which attacks a class u. Consider the following experiment:

```
*Experiment* $\mathbf{Exp}^{\mathtt{STRONG-REC}}_{\mathtt{STAT}_u}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $keys_u \leftarrow Keys_u(k)$
    $k'_u \leftarrow \mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u)$
    **return** $k'_u$
```

The advantage of $\mathtt{STAT}_u$ is defined as

$$\mathbf{Adv}^{\mathtt{STRONG-REC}}_{\mathtt{STAT}_u}(1^\tau, G) = Pr[k'_u = k_u].$$

The scheme is said to be *secure in the sense of* STRONG-REC-ST if, for each graph $G = (V, E)$ in Γ and each class $u \in V$, the function $\mathbf{Adv}^{\mathtt{STRONG-REC}}_{\mathtt{STAT}_u}(1^\tau, G)$ is negligible, for each static adversary $\mathtt{STAT}_u$ whose time complexity is polynomial in τ.

IV. Implications and Separations

In this section, we analyze the relations between the security definitions described in Section III. In particular, we show implications and separations occurring between such notions. Figure 1 summarizes our results.

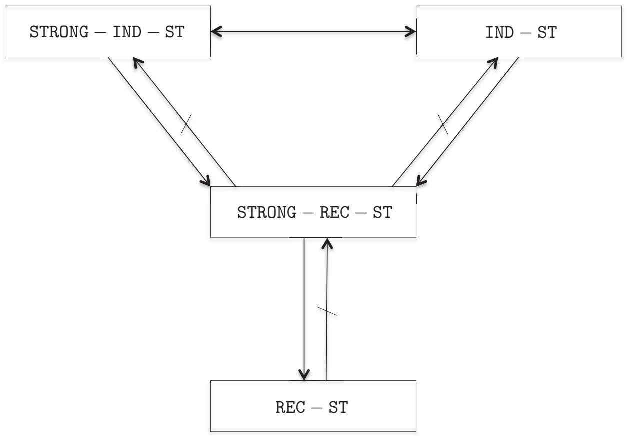

Figure 1: Relations between the security notions for hierarchical key assignment schemes.

It is easy to see that any adversary which breaks the security of the key assignment scheme in the sense of

STRONG-IND-ST can be easily turned into another adversary which breaks the security of the key assignment scheme in the sense of STRONG-REC-ST. Hence, the next result holds.

Theorem 4.1: [STRONG-IND-ST$\Rightarrow$STRONG-REC-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If a hierarchical key assignment scheme for Γ is secure in the sense of STRONG-IND-ST, then it is also secure in the sense of STRONG-REC-ST.

In the following, we show that security against strong key recovery does not necessarily imply security with respect to strong key indistinguishability. Let (Gen, Der) be a hierarchical key assignment scheme which is secure in the sense of STRONG-REC-ST. We construct another scheme (Gen', Der') and we show that it is secure in the sense of STRONG-REC-ST but is not secure in the sense of STRONG-IND-ST. Let $u \in V$ be a class and let k_u be the key assigned by Gen to u. Algorithm Gen' computes the key assigned to class u as $k'_u = 1||k_u$, where the symbol $||$ denotes string concatenation. All other values computed by Gen' are exactly the same as the ones computed by Gen. Algorithm Der' first computes k_u by using Der, then obtains $k'_u = 1||k_u$. Let $\mathtt{STAT_u}$ be a static adversary that simply checks whether the first bit x_0 of the challenge x, corresponding either to the key k'_u or to a random string having the same length as k'_u, is equal to 0. If this happens, then $\mathtt{STAT_u}$ can easily conclude that the challenge x does not correspond to the key k'_u, but is a random string. Since the advantage $\mathbf{Adv}^{\mathtt{STRONG-IND}}_{\mathtt{STAT}_u}$ is non-negligible, it follows that (Gen', Der') is not secure in the sense of STRONG-IND-ST. On the other hand, (Gen', Der') is secure in the sense of STRONG-REC-ST. Assume by contradiction that (Gen', Der') is not secure in the sense of STRONG-REC-ST. It follows that also (Gen, Der) is not secure in the sense of STRONG-REC-ST, thus leading to a contradiction. For this reason, the next result holds.

Theorem 4.2: [STRONG-REC-ST$\not\Rightarrow$STRONG-IND-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If there exists a hierarchical key assignment scheme for Γ which is secure in the sense of STRONG-REC-ST, then there exists a hierarchical key assignment scheme for Γ that is secure in the sense of STRONG-REC-ST but which is not secure in the sense of STRONG-IND-ST.

The relations between the definitions of security against strong key recovery and security against key recovery have been established by Freire et al. [41]. In particular, they showed that the two notions of security against key recovery and against strong key recovery *are separated*, i.e., there exist hierarchical key assignment schemes that are secure against key recovery but which are not secure against strong key recovery. An example of such schemes is the one based on pseudorandom functions, proposed by Atallah et al. [14]. Thus, the following theorems hold.

Theorem 4.3: [STRONG-REC-ST$\Rightarrow$REC-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If a hierarchical key assignment scheme for Γ is secure in the sense of STRONG-REC-ST, then it is also secure in the sense of REC-ST.

Theorem 4.4: [REC-ST$\not\Rightarrow$STRONG-REC-ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If there exists a hierarchical key assignment scheme for Γ which is secure in the sense of REC−ST, then there exists a hierarchical key assignment scheme for Γ that is secure in the sense of REC−ST but which is not secure in the sense of STRONG−REC−ST.

However, Freire et al. [41] did not clarify the relations between the notions of security with respect to key indistinguishability and with respect to strong key indistinguishability. As stated by the next theorem, it is easy to see that security with respect to strong key indistinguishability implies security with respect to key indistinguishability. However, nothing is known about the other direction.

Theorem 4.5: [STRONG−IND−ST⇒IND−ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If a hierarchical key assignment scheme for Γ is secure in the sense of STRONG−IND−ST, then it is also secure in the sense of IND−ST.

In the following, we show that security with respect to strong key indistinguishability is *not stronger* than the one with respect to key indistinguishability, that is to say, STRONG−IND−ST and IND−ST are (*polynomially*) equivalent.

Theorem 4.6: [IND−ST⇒STRONG−IND−ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If a hierarchical key assignment scheme for Γ is secure in the sense of IND−ST, then it is also secure in the sense of STRONG−IND−ST.

Proof: Assume by contradiction that there exists a hierarchical key assignment scheme Σ for a graph $G = (V, E)$ in Γ, which is secure in the sense of IND−ST but that is not secure in the sense of STRONG−IND−ST. Therefore, there exists a class $u \in V$ and a static adversary $\mathtt{STAT}_u$ which is able to distinguish between experiments $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$ with non-negligible advantage. Recall that the only difference between $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$ is the last input of $\mathtt{STAT}_u$, which corresponds to a random value chosen in $\{0, 1\}^{\tau}$ in the first experiment and to the real key k_u in the second one.

Let $P_u = \{v \in V \setminus \{u\} : u \in A_v\}$ be the set of predecessors of class u, and, w.l.o.g., let $(u_1, \ldots, u_m)$ be any topological ordering of its elements. Notice that the sequence $keys_u$, which is an input of $\mathtt{STAT}_u$ in both the experiments $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$, contains exactly the keys $k_{u_1}, \ldots, k_{u_m}$. First of all, it is easy to observe that if $m = 0$ the sequence $keys_u$ is empty, thus the experiments $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$ correspond to $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{IND-1}}$, respectively. In this case, since $\mathtt{STAT}_u$ is able to distinguish between such experiments with non-negligible advantage, it follows that the scheme Σ is not secure in the sense of IND−ST, which is a contradiction.

In addition, consider the case in which $m > 0$. We will show how to turn the adversary $\mathtt{STAT}_u$ into another polynomial-time adversary $\mathtt{STAT'}_{u_h}$, where $u_h \in P_u$, which breaks the scheme Σ in the sense of IND−ST, thus leading to a contradiction. We construct two sequences $\mathbf{Exp}_u^{1,1}, \ldots, \mathbf{Exp}_u^{1,m+1}$ and $\mathbf{Exp}_u^{2,1}, \ldots, \mathbf{Exp}_u^{2,m+1}$ of $m + 1$ experiments each, all defined over the same probability space, where the first experiment of the former sequence, that is $\mathbf{Exp}_u^{1,1}$, is equal to $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$, whereas, the last experiment of the latter sequence, that is $\mathbf{Exp}_u^{2,m+1}$, is equal to $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$.

For any $q = 2, \ldots, m+1$, experiment $\mathbf{Exp}_u^{1,q}$ in the first sequence is defined as follows:

```
Experiment $\mathbf{Exp}_u^{1,q}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $keys_u^q \leftarrow Keys_u^q(k)$
    $d \leftarrow \mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u^q, \rho)$
    return $d$
```

The output of the algorithm $Keys_u^q$ is the sequence $keys_u^q$ where the first $q-1$ values are independently chosen at random in $\{0,1\}^\tau$ and, if $q \leq m$, the other $m-q+1$ values correspond to the keys assigned to the classes $u_q, \ldots, u_m$.

On the other hand, for any $q = 1, \ldots, m$, experiment $\mathbf{Exp}_u^{2,q}$ in the second sequence is defined as follows:

```
Experiment $\mathbf{Exp}_u^{2,q}(1^\tau, G)$
    $(s, k, pub) \leftarrow Gen(1^\tau, G)$
    $corr_u \leftarrow Corrupt_u(s)$
    $keys_u^{m-q+2} \leftarrow Keys_u^{m-q+2}(k)$
    $d \leftarrow \mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u^{m-q+2}, k_u)$
    return $d$
```

where $keys_u^{m-q+2}$ denotes the sequence where the first $m-q+1$ values are independently chosen at random in $\{0,1\}^\tau$ and, if $q \geq 2$, the other $q-1$ values correspond to the keys assigned to the classes $u_{m-q+2}, \ldots, u_m$. Since $\mathbf{Exp}_u^{1,1}$, which corresponds to $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-0}}$, and $\mathbf{Exp}_u^{2,m+1}$, which corresponds to $\mathbf{Exp}_{\mathtt{STAT}_u}^{\mathtt{STRONG-IND-1}}$, can be distinguished by $\mathtt{STAT}_u$ with non-negligible advantage, then there exists at least a pair of consecutive experiments, in the sequence of $2m+2$ experiments obtained by composition of the two above defined sequences, which are distinguishable by $\mathtt{STAT}_u$ with non-negligible advantage.

We first show that such a pair cannot consist of the two extremal experiments, namely, the last experiment of the first sequence, that is $\mathbf{Exp}_u^{1,m+1}$, and the first experiment of the second sequence, that is $\mathbf{Exp}_u^{2,1}$. Assume by contradiction that $\mathtt{STAT}_u$ is able to distinguish between $\mathbf{Exp}_u^{1,m+1}$ and $\mathbf{Exp}_u^{2,1}$ with non-negligible advantage. Notice that the only difference between such two experiments is the last input of $\mathtt{STAT}_u$, which corresponds to a random value chosen in $\{0,1\}^\tau$ in experiment $\mathbf{Exp}_u^{1,m+1}$, and to the real key k_u in experiment $\mathbf{Exp}_u^{2,1}$. We show how to construct another adversary $\mathtt{STAT'}_u$ which breaks the security of the scheme Σ in the sense of $\mathtt{IND-ST}$, by using the adversary $\mathtt{STAT}_u$. The adversary $\mathtt{STAT'}_u$, on inputs 1^τ, G, the sequence of private information $corr_u$ and a final value α, corresponding either to the key k_u or to a random value chosen in $\{0,1\}^\tau$, constructs the sequence $keys_u^{m+1}$ needed for $\mathtt{STAT}_u$ choosing independently at random m elements in $\{0,1\}^\tau$. Then, $\mathtt{STAT'}_u$ outputs the same output as $\mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u^{m+1}, \alpha)$. Clearly, since $\mathtt{STAT}_u$ is able to distinguish between $\mathbf{Exp}_u^{1,m+1}$

and $\mathbf{Exp}_u^{2,1}$ with non-negligible advantage, then $\mathtt{STAT'}_u$ is able to distinguish between $\mathbf{Exp}_{\mathtt{STAT'}_u}^{\mathtt{IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT'}_u}^{\mathtt{IND-1}}$ with non-negligible advantage, thus breaking the security of the scheme Σ in the sense of IND−ST. Contradiction. Thus, the pair of consecutive experiments which can be distinguished by $\mathtt{STAT}_u$, belongs either to the first sequence or to the second one.

Assume that the pair of distinguishable consecutive experiments belongs to the first sequence and it is composed by $\mathbf{Exp}_u^{1,h}$ and $\mathbf{Exp}_u^{1,h+1}$, for some $h = 1, \ldots, m$. Notice that the views of $\mathtt{STAT}_u$ in such two consecutive experiments differ only for a single value, which corresponds to the key k_{u_h} in $\mathbf{Exp}_u^{1,h}$ and to a random value chosen in $\{0,1\}^\tau$ in $\mathbf{Exp}_u^{1,h+1}$. We show how to construct an adversary $\mathtt{STAT''}_{u_h}$ which breaks the security of the scheme Σ in the sense of IND−ST, by using the adversary $\mathtt{STAT}_u$. In particular, we show that $\mathtt{STAT''}_{u_h}$ is able to distinguish between experiments $\mathbf{Exp}_{\mathtt{STAT''}_{u_h}}^{\mathtt{IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT''}_{u_h}}^{\mathtt{IND-1}}$ with non-negligible advantage. The adversary $\mathtt{STAT''}_{u_h}$, on inputs 1^τ, G, the sequence of private information $corr_{u_h}$ and a final value α, corresponding either to the key k_{u_h} or to a random value chosen in $\{0,1\}^\tau$, constructs the inputs for $\mathtt{STAT}_u$ as follows:

- First, $\mathtt{STAT''}_{u_h}$ extracts from $corr_{u_h}$ the sequence $corr_u$. This can be done since $u_h \in P_u$, i.e., u_h is a predecessor of u, hence classes which are corrupted for u are also corrupted for u_h and their private information is in $corr_{u_h}$.
- Then, $\mathtt{STAT''}_{u_h}$ uses $corr_{u_h}$ and α to construct a sequence $keys_u^\alpha$, which corresponds either to $keys_u^h$ or to $keys_u^{h+1}$. In particular, the first $h-1$ elements of $keys_u^\alpha$ are independently chosen at random in $\{0,1\}^\tau$, the h-th element is set equal to α, whereas, the remaining $m-h+1$ ones, which correspond to the keys of the classes $u_{h+1}, \ldots, u_m$, are computed by using the private information of such classes, which are contained in $corr_{u_h}$.
- Moreover, the last input for $\mathtt{STAT}_u$ is set equal to a random value ρ chosen in $\{0,1\}^\tau$.

Finally, $\mathtt{STAT''}_{u_h}$ outputs the same output as $\mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u^\alpha, \rho)$. Clearly, since $\mathtt{STAT}_u$ is able to distinguish between $\mathbf{Exp}_u^{1,h}$ and $\mathbf{Exp}_u^{1,h+1}$ with non-negligible advantage, then $\mathtt{STAT''}_{u_h}$ is able to distinguish between $\mathbf{Exp}_{\mathtt{STAT''}_{u_h}}^{\mathtt{IND-0}}$ and $\mathbf{Exp}_{\mathtt{STAT''}_{u_h}}^{\mathtt{IND-1}}$ with non-negligible advantage, thus breaking the security of the scheme Σ in the sense of IND−ST. Contradiction.

Notice that if the pair of distinguishable consecutive experiments belongs to the second sequence, i.e., is composed by $\mathbf{Exp}_u^{2,h}$ and $\mathbf{Exp}_u^{2,h+1}$, for some $h = 1, \ldots, m$, the proof is similar to the previous case. ■

From Theorems 4.6, 4.1 and 4.3 we obtain the next result, which has already been proved in [32].

Theorem 4.7: [IND−ST⇒REC−ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies. If a hierarchical key assignment scheme for Γ is secure in the sense of IND−ST, then it is also secure in the sense of REC−ST.

On the other hand, from Theorems 4.4, 4.2, 4.5, and 4.6, the next result, which has already been proved in [32], follows.

Theorem 4.8: [REC−ST⇏IND−ST] Let Γ be a family of graphs corresponding to partially ordered hierarchies.

If there exists a hierarchical key assignment scheme for Γ which is secure in the sense of `REC-ST`, then there exists a hierarchical key assignment scheme for Γ that is secure in the sense of `REC-ST` but which is not secure in the sense of `IND-ST`.

V. Towards Security against Strong Key Recovery

As said in the previous section, the two notions of security against key recovery and against strong key recovery *are separated*, i.e., there exist hierarchical key assignment schemes that are secure against key recovery but which are not secure against strong key recovery. In this section, we investigate the possibility of obtaining a scheme which is secure with respect to the stronger notion, starting from any scheme which is secure with respect to the weaker one.

The idea behind our construction is the following. Given a graph $G = (V, E)$ representing a partially ordered hierarchy, we construct another graph G' which represents the same hierarchy, but that has $|V|$ *additional* classes. Then, we use a hierarchical key assignment scheme to assign private information and encryption keys to the classes of G'. This assignment can be easily turned into an assignment for the original graph G. Indeed, the private information for each class in G is set equal to that assigned to the same class in G', whereas, the encryption keys for classes in G are those assigned to the additional classes in G'. We will show how the resulting hierarchical key assignment scheme for G satisfies security agains strong key recovery, provided that the underlying scheme for G' satisfies security against key recovery.

Formally, let Γ be a family of graphs corresponding to partially ordered hierarchies. For each graph $G = (V, E)$ in Γ we define a graph transformation, whose output, denoted by $G' = (V', E')$, is called the *extended graph for* G. We denote by Γ' the family of extended graphs for elements in Γ. The transformation works as follows:

- For each $u \in V$, we place two classes u and u_0 in V';
- For each class $u \in V$, we place the edge (u, u_0) in E';
- For each $(u, v) \in E$, we place the edge (u, v) in E'.

Figure 2 shows an example of the extended graph for $G = (V, E)$, where $V = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (b, d), (c, d)\}$.

Let Γ' be the family of extended graphs for elements in Γ and let (Gen', Der') be a hierarchical key assignment scheme for Γ'. The proposed key assignment scheme for Γ works as follows.

Algorithm $Gen(1^\tau, G)$

```
1) Construct the extended graph $G' = (V', E')$ for $G = (V, E)$;
2) Let $(s', k', pub')$ be the output of $Gen'$ on inputs $(1^\tau, G')$;
3) For each class $u \in V$, let $s_u = s'_u$;
4) For each class $u \in V$, let $k_u = k'_{u_0}$;
5) Let $s$ and $k$ be the sequences of private information and keys, respectively, computed in the previous steps;
6) Output $(s, k, pub)$.
```

Algorithm $Der(1^\tau, G, u, v, s_u, pub)$

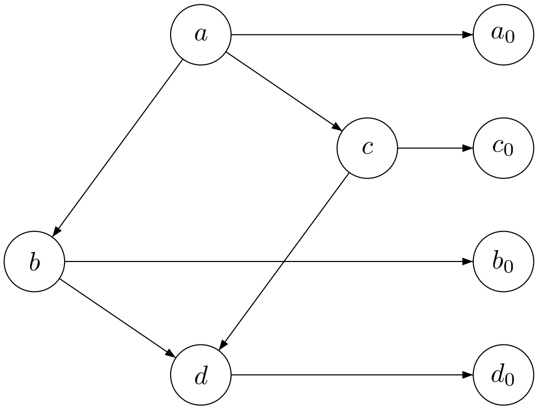

Figure 2: The graph $G' = (V', E')$, where $V = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (b, d), (c, d)\}$.

1) Let k'_{v_0} be the output of Der' on inputs $(1^\tau, G', u, v_0, s'_u, pub')$;
2) Output $k_v = k'_{v_0}$.

The next theorem states that if (Gen', Der') is secure against key recovery, then (Gen, Der) is secure against strong key recovery.

Theorem 5.1: If (Gen', Der') is secure in the sense of REC−ST, then (Gen, Der) is secure in the sense of STRONG-REC−ST.

Proof: Assume by contradiction that the scheme (Gen, Der) is not secure in the sense of STRONG-REC−ST. Therefore, there exists a graph $G = (V, E)$ in Γ and a class $u \in V$ for which there exists a polynomial time adversary $\mathtt{STAT}_u$ whose advantage $\mathbf{Adv}^{\mathtt{STRONG-REC}}_{\mathtt{STAT}_u}(1^\tau, G)$ is non-negligible. We show how to construct a polynomial-time adversary which, by using $\mathtt{STAT}_u$, is able to break the security of the scheme (Gen', Der') in the sense of REC−ST. Such an adversary, which we denote by $\mathtt{STAT'}_{u_0}$, on inputs 1^τ, an extended graph G', the public information pub', and the sequence $corr'_{u_0}$ of private information held by corrupted users, constructs the inputs for $\mathtt{STAT}_u$ as follows:

- First, $\mathtt{STAT'}_{u_0}$ constructs the graph G from G', so that G' is the extended graph for G. This operation simply involves the cancellation of all the classes $v_0 \in V'$.
- Then, the adversary sets the public information pub to be equal to pub'.
- Afterwards, the adversary extracts the sequence $corr_u$ from $corr'_{u_0}$. Indeed, $corr'_{u_0}$ contains the private information s'_v for each class $v \in F_u$.
- Moreover, the adversary constructs the sequence $keys_u$ as follows: first, it extracts from the sequence $corr'_{u_0}$ the private information s'_{v_0} for each $v \neq u$. Such values are then used to compute the sequence of keys k'_{v_0} for each $v \neq u$. These values are exactly the elements of the sequence $keys_u$.

Finally, $\mathtt{STAT'}_{u_0}$ returns the same output as $\mathtt{STAT}_u(1^\tau, G, pub, corr_u, keys_u)$. Therefore, it is easy to see that

$$\mathbf{Adv}^{\texttt{REC}}_{\texttt{STAT}'_{u_0}}(1^\tau, G') = \mathbf{Adv}^{\texttt{STRONG-REC}}_{\texttt{STAT}_u}(1^\tau, G).$$

Since $\mathbf{Adv}^{\texttt{STRONG-REC}}_{\texttt{STAT}_u}(1^\tau, G)$ is non-negligible, it follows that the adversary $\texttt{STAT}'_{u_0}$ is able to break the security of the scheme (Gen', Der') in the sense of REC−ST. Contradiction. ■

VI. Conclusions

In this paper we have explored the relations between all security notions for hierarchical key assignment schemes and, in particular, we have shown that security with respect to strong key indistinguishability is *not stronger* than the one with respect to key indistinguishability. We have also proposed a general construction yielding a hierarchical key assignment scheme offering security against strong key recovery, given any hierarchical key assignment scheme which guarantees security against key recovery.

VII. Acknowledgements

We would like to thank Kenny Paterson for his valuable and detailed comments which helped us to improve the readability of this paper.

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," *ACM Trans. Comput. Syst.*, vol. 1, no. 3, pp. 239–248, 1983. [Online]. Available: http://doi.acm.org/10.1145/357369.357372

[2] S. J. MacKinnon, P. D. Taylor, H. Meijer, and S. G. Akl, "An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy," *IEEE Trans. Computers*, vol. 34, no. 9, pp. 797–802, 1985. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TC.1985.1676635

[3] L. Harn and H. Lin, "A Cryptographic Key Generation Scheme for Multilevel Data Security," *Computers & Security*, vol. 9, no. 6, pp. 539–546, 1990. [Online]. Available: http://dx.doi.org/10.1016/0167-4048(90)90132-D

[4] H. Liaw, S. Wang, and C. Lei, "A Dynamic Cryptographic Key Assignment Scheme in a Tree Structure," *Computers & Mathematics with Applications*, vol. 25, no. 6, pp. 109 – 114, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/089812219390305F

[5] H. Min-Shiang, "A Cryptographic Key Assignment Scheme in a Hierarchy for Access Control," *Math. Comput. Model.*, vol. 26, no. 2, pp. 27–31, Jul. 1997. [Online]. Available: http://dx.doi.org/10.1016/S0895-7177(97)00120-9

[6] C.-H. Lin, "Dynamic Key Management Schemes for Access Control in a Hierarchy," *Computer Communications*, vol. 20, no. 15, pp. 1381 – 1385, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014036649700100X

[7] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," *Inf. Process. Lett.*, vol. 27, no. 2, pp. 95–98, 1988. [Online]. Available: http://dx.doi.org/10.1016/0020-0190(88)90099-3

[8] T. Wu and C. Chang, "Cryptographic Key Assignment Scheme for Hierarchical Access Control," *Comput. Syst. Sci. Eng.*, vol. 16, no. 1, pp. 25–28, 2001.

[9] T. Chen and Y. Chung, "Hierarchical Access Control Based on Chinese Remainder Theorem and Symmetric Algorithm," *Computers & Security*, vol. 21, no. 6, pp. 565–570, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-4048(02)01016-7

[10] V. R. L. Shen and T. Chen, "A Novel Key Management Scheme Based on Discrete Logarithms and Polynomial Interpolations," *Computers & Security*, vol. 21, no. 2, pp. 164–171, 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-4048(02)00211-0

[11] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and Efficient Key Management for Access Hierarchies," in *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*, V. Atluri, C. Meadows, and A. Juels, Eds. ACM, 2005, pp. 190–202. [Online]. Available: http://doi.acm.org/10.1145/1102120.1102147

[12] A. D. Santis, A. L. Ferrara, and B. Masucci, "Efficient Provably-Secure Hierarchical Key Assignment Schemes," in *Mathematical Foundations of Computer Science 2007, 32nd International Symposium, MFCS 2007, Ceský Krumlov, Czech Republic, August 26-31, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Kucera and A. Kucera, Eds., vol. 4708. Springer, 2007, pp. 371–382. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74456-6_34

[13] M. J. Atallah, M. Blanton, and K. B. Frikken, "Key Management for Non-Tree Access Hierarchies," in *SACMAT 2006,11th ACM Symposium on Access Control Models and Technologies, Lake Tahoe, California, USA, June 7-9, 2006, Proceedings*, D. F. Ferraiolo and I. Ray, Eds. ACM, 2006, pp. 11–18. [Online]. Available: http://doi.acm.org/10.1145/1133058.1133062

[14] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 3, 2009. [Online]. Available: http://doi.acm.org/10.1145/1455526.1455531

[15] P. D'Arco, A. D. Santis, A. L. Ferrara, and B. Masucci, "Security and Tradeoffs of the Akl-Taylor Scheme and Its Variants," in *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings*, ser. Lecture Notes in Computer Science, R. Královic and D. Niwinski, Eds., vol. 5734. Springer, 2009, pp. 247–257. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03816-7_22

[16] ——, "Variations on a theme by Akl and Taylor: Security and Tradeoffs," *Theor. Comput. Sci.*, vol. 411, no. 1, pp. 213–227, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2009.09.028

[17] A. D. Santis, A. L. Ferrara, and B. Masucci, "Efficient Provably-Secure Hierarchical Key Assignment Schemes," *Theor. Comput. Sci.*, vol. 412, no. 41, pp. 5684–5699, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2011.06.024

[18] E. S. V. Freire and K. G. Paterson, "Provably Secure Key Assignment Schemes from Factoring," in *Information Security and Privacy - 16th Australasian Conference, ACISP 2011, Melbourne, Australia, July 11-13, 2011. Proceedings*, ser. Lecture Notes in Computer Science, U. Parampalli and P. Hawkes, Eds., vol. 6812. Springer, 2011, pp. 292–309. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22497-3_19

[19] J. Yeh, R. Chow, and R. Newman, "A Key Assignment for Enforcing Access Control Policy Exceptions," in *Proc. of the International Symposium on Internet Technology*, 1998, pp. 54–59.

[20] I.-C. Lin, M.-S. Hwang, and C.-C. Chang, "A New Key Assignment Scheme for Enforcing Complicated Access Control Policies in Hierarchy," *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457 – 462, 2003, selected papers from the IEEE/ACM International Symposium on Cluster Computing and the Grid, Berlin-Brandenburg Academy of Sciences and Humanities, Berlin, Germany, 21-24 May 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X02002005

[21] A. D. Santis, A. L. Ferrara, and B. Masucci, "Cryptographic Key Assignment Schemes for Any Access Control Policy," *Inf. Process. Lett.*, vol. 92, no. 4, pp. 199–205, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.ipl.2004.03.019

[22] ——, "Unconditionally Secure Key Assignment Schemes," *Discrete Applied Mathematics*, vol. 154, no. 2, pp. 234–252, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2005.03.025

[23] W. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 182–188, 2002. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/69.979981

[24] H. Chien, "Efficient Time-Bound Hierarchical Key Assignment Scheme," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 10, pp. 1301–1304, 2004. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2004.59

[25] H. Huang and C. Chang, "A New Cryptographic Key Assignment Scheme with Time-Constraint Access Control in a Hierarchy," *Computer Standards & Interfaces*, vol. 26, no. 3, pp. 159–166, 2004. [Online]. Available: http://dx.doi.org/10.1016/S0920-5489(03)00073-4

[26] J. Yeh, "An RSA-based Time-Bound Hierarchical Key Assignment Scheme for Electronic Article Subscription," in *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, O. Herzog, H. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, Eds. ACM, 2005, pp. 285–286. [Online]. Available: http://doi.acm.org/10.1145/1099554.1099629

[27] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, 2006, pp. 288–297. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180441

[28] S. Wang and C. Laih, "Merging: An Efficient Solution for a Time-Bound Hierarchical Key Assignment Scheme," *IEEE Trans. Dependable Sec. Comput.*, vol. 3, no. 1, pp. 91–100, 2006. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TDSC.2006.15

[29] W. Tzeng, "A Secure System for Data Access Based on Anonymous Authentication and Time-Dependent Hierarchical Keys," in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006, Taipei, Taiwan, March 21-24, 2006*, F. Lin, D. Lee, B. P. Lin, S. Shieh, and S. Jajodia, Eds. ACM, 2006, pp. 223–230. [Online]. Available: http://doi.acm.org/10.1145/1128817.1128851

[30] A. D. Santis, A. L. Ferrara, and B. Masucci, "New Constructions for Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," in *SACMAT 2007, 12th ACM Symposium on Access Control Models and Technologies, Sophia Antipolis, France, June 20-22, 2007, Proceedings*, V. Lotz and B. M. Thuraisingham, Eds. ACM, 2007, pp. 133–138. [Online]. Available: http://doi.acm.org/10.1145/1266840.1266861

[31] ——, "New Constructions for Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," *Theor. Comput. Sci.*, vol. 407, no. 1-3, pp. 213–230, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2008.05.021

[32] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," *J. Cryptology*, vol. 25, no. 2, pp. 243–270, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00145-010-9094-6

[33] X. Yi and Y. Ye, "Security of Tzeng's Time-Bound Key Assignment Scheme for Access Control in a Hierarchy," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 1054–1055, 2003. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2003.1209023

[34] X. Yi, "Security of Chien's Efficient Time-Bound Hierarchical Key Assignment Scheme," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1298–1299, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2005.152

[35] Q. Tang and C. J. Mitchell, "Comments On a Cryptographic Key Assignment Scheme," *Computer Standards & Interfaces*, vol. 27, no. 3, pp. 323–326, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2004.07.001

[36] A. D. Santis, A. L. Ferrara, and B. Masucci, "Enforcing the Security of a Time-Bound Hierarchical Key Assignment Scheme," *Inf. Sci.*, vol. 176, no. 12, pp. 1684–1694, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2005.07.002

[37] S. Goldwasser and S. Micali, "Probabilistic Encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984. [Online]. Available: http://dx.doi.org/10.1016/0022-0000(84)90070-9

[38] M. J. Atallah, M. Blanton, and K. B. Frikken, "Incorporating Temporal Capabilities in Existing Key Management Schemes," in *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, ser. Lecture Notes in Computer Science, J. Biskup and J. Lopez, Eds., vol. 4734. Springer, 2007, pp. 515–530. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74835-9_34

[39] E. Bertino, N. Shang, and S. S. W. Jr., "An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting," *IEEE Trans. Dependable Sec. Comput.*, vol. 5, no. 2, pp. 65–70, 2008. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TDSC.2007.70241

[40] A. Castiglione, A. De Santis, and B. Masucci, "Hierarchical and Shared Key Assignment," in *17th International Conference on Network-Based Information Systems, NBIS 2014, IEEE*, 2014, pp. 263–270. [Online]. Available: http://dx.doi.org/10.1109/NBiS.2014.106

[41] E. S. V. Freire, K. G. Paterson, and B. Poettering, "Simple, Efficient and Strongly KI-Secure Hierarchical Key Assignment Schemes," in *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, ser. Lecture Notes in Computer Science, E. Dawson, Ed., vol. 7779. Springer, 2013, pp. 101–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36095-4_7

[42] M. Cafaro, R. Civino, and B. Masucci, "On the Equivalence of Two Security Notions for Hierarchical Key Assignment Schemes in the Unconditional Setting," *IEEE Trans. Dependable Sec. Comput.*, 2014. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2014.2355841

[43] A. V. Aho, M. R. Garey, and J. D. Ullman, "The Transitive Reduction of a Directed Graph," *SIAM J. Comput.*, vol. 1, no. 2, pp. 131–137, 1972. [Online]. Available: http://dx.doi.org/10.1137/0201008

[44] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, "A Note on Time-Bound Hierarchical Key Assignment Schemes," *Inf. Process. Lett.*, vol. 113, no. 5-6, pp. 151–155, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ipl.2013.01.006

[45] S. Goldwasser, S. Micali, and R. L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988. [Online]. Available: http://dx.doi.org/10.1137/0217017