Variations on a theme by Akl and Taylor: Security and tradeoffs∗

Paolo D’Arco, Alfredo De Santis, Anna Lisa Ferrara, Barbara Masucci *

Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084 Fisciano, SA, Italy

ARTICLE INFO

Article history:
Received 1 February 2009
Received in revised form 9 September 2009
Accepted 20 September 2009
Communicated by A. Fiat

Keywords:
Access control
Key assignment
Provable security
Efficiency

ABSTRACT

In 1983, Akl and Taylor [Cryptographic solution to a problem of access control in a hierarchy, ACM Transactions on Computer Systems 1 (3) (1983) 239–248] first suggested the use of cryptographic techniques to enforce access control in hierarchical structures. Due to its simplicity and versatility, the scheme has been used, for more than twenty years, to implement access control in several different domains, including mobile agent environments and XML documents. However, despite its use over time, the scheme has never been fully analyzed with respect to security and efficiency requirements. In this paper we provide new results on the Akl–Taylor scheme and its variants. More precisely:

• We provide a rigorous analysis of the Akl–Taylor scheme. We consider different key assignment strategies and prove that the corresponding schemes are secure against key recovery.
• We show how to obtain different tradeoffs between the amount of public information and the number of steps required to perform key derivation in the proposed schemes.
• We also look at the MacKinnon et al. and Harn and Lin schemes and prove they are secure against key recovery.
• We describe an Akl–Taylor based key assignment scheme with time-dependent constraints and prove the scheme efficient, flexible and secure.
• We propose a general construction, which is of independent interest, yielding a key assignment scheme offering security w.r.t. key indistinguishability, given any key assignment scheme which guarantees security against key recovery.
• Finally, we show how to use our construction, along with our assignment strategies and tradeoffs, to obtain an Akl–Taylor scheme, secure w.r.t. key indistinguishability, requiring a constant amount of public information.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Hierarchical key assignment schemes are widely used to implement secure access control policies in applications where users and resources can be modeled through partially ordered hierarchies. Since hierarchies are a natural way of organizing users according to their positions as parts of an organization, they are widely employed in many different application areas (e.g. database management systems, computer networks, operating systems, military, government communications) and play a fundamental role within any scenario which can be modeled by using Role Based Access Control (RBAC).


∗ Corresponding author.

E-mail addresses: paodar@dia.unisa.it (P. D’Arco), ads@dia.unisa.it (A. De Santis), ferrara@dia.unisa.it (A.L. Ferrara), masucci@dia.unisa.it (B. Masucci).
In 1983, Akl and Taylor [1] suggested the use of cryptographic techniques to enforce access control in hierarchical structures. In particular, they designed a hierarchical key assignment scheme where each class is assigned an encryption key that can be used, along with some public parameters, to compute the key assigned to all classes lower down in the hierarchy. Due to its simplicity and versatility, the Akl–Taylor scheme has been widely used to enforce access control in several different domains, including mobile agent environments [16] and XML documents [6]. Moreover, it has also been used as a starting point in the design of key assignment schemes for the enforcement of more general access control policies, including those with transitive and anti-symmetrical exceptions [22,7], as well as those with time-dependent constraints [21,23]. Furthermore, Asano [2] and Attrapadung and Kobara [5] have used the Akl–Taylor scheme in the context of broadcast encryption.

Akl and Taylor related the security of their scheme to the infeasibility of extracting \( r \)th roots modulo \( n \), where \( r > 1 \) is an integer and \( n \) is the product of two large unknown primes. However, their analysis gives only an intuition for the security of the scheme. Later on, Goldwasser and Micali [12], introduced the use of security reductions to provide rigorous security arguments for cryptographic protocols. Security reductions aim at reducing the security of a protocol to the security of a presumed hard computational problem for which no efficient (i.e., probabilistic polynomial time) solving algorithm is known. Despite its use for many years, there has been no attempt to fully analyze the security of the Akl–Taylor scheme according to the Goldwasser–Micali paradigm. The issue of providing rigorous security proofs is very important since many key assignment schemes have been shown insecure against collusive attacks (see [4] for references).

Atallah et al. [3] first proposed two different notions of security for hierarchical key assignment schemes: security against key recovery and with respect to key indistinguishability. In the key recovery case, an adversary cannot compute a key which cannot be derived by the users he has corrupted; whereas, in the key indistinguishability case, the adversary is not even able to distinguish the key from a random string of the same length. Hierarchical key assignment schemes satisfying the above notions of security have been proposed in [3,4,8].

1.1. Our contribution

In this paper we analyze the Akl–Taylor scheme as well as some of its variants with respect to security and efficiency requirements.

Security. We analyze the Akl–Taylor scheme according to the definitions in [3]. We carefully specify how to choose the public parameters in order to get instances of the scheme which are secure against key recovery under the RSA assumption.

In the Akl–Taylor scheme the complexity of the key derivation increases with the number of classes in the hierarchy. In order to speed up the key derivation, MacKinnon et al. [17] described a different generation of public values, whereas, Harn and Lin [14] proposed a variant of the Akl–Taylor scheme which provides a more efficient key derivation for broad and shallow hierarchies. We show that the MacKinnon et al. and the Harn–Lin variants are secure against key recovery.

Similar considerations allow us to prove the security of the Akl–Taylor scheme with a different public values assignment that we call the reduced Akl–Taylor assignment. Schemes using the reduced Akl–Taylor assignment have been used for the enforcement of more general access control policies, including those with transitive and anti-symmetrical exceptions [7,22], as well as those with time-dependent constraints [21,23]. Thus, our arguments also yield formal proofs for schemes in [7,21,23].

We also consider the issue of designing an Akl–Taylor based scheme offering security with respect to key indistinguishability. Specifically, we propose a general construction, which is of independent interest, yielding a key assignment scheme offering security with respect to key indistinguishability, given any key assignment scheme which guarantees security against key recovery.

Efficiency. The considered Akl–Taylor scheme is still secure when only part of the public information as small as a single prime number is published. This at the cost of a more expensive key derivation. Thus, we show a tradeoff between the size of the public information and the complexity of the key derivation.

We point out that Akl–Taylor based schemes, compared with other existing provably-secure schemes, provide better performances except for the key derivation complexity. In particular, no other provably-secure scheme with time constraints requires as little public information as that required by time-dependent Akl–Taylor based schemes.

2. The model and definitions

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 \( \leq \) that partially orders the set of classes \( V \) is defined in accordance with authority, position, or power of each class in \( V \). The poset \( (V, \leq) \) is called a partially ordered hierarchy. For any two classes \( u \) and \( v \), the notation \( v \leq u \) is used to indicate that the users in \( u \) can access \( v \)'s data. Clearly, since \( u \) can access its own data, it holds that \( u \leq u \), for any \( u \in V \). We denote by \( A_u \) the set of nodes to whom node \( u \) has access to, i.e., \( A_u = \{ v \in V : v \leq u \} \), for any \( u \in V \). The partially ordered hierarchy \( (V, \leq) \) 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 \( u \) to class \( v \) if and only if \( u \leq v \). We denote by \( G = (V, E) \) the minimal representation of the graph \( G^* \), that is, the directed acyclic graph corresponding to the transitive and reflexive reduction of the graph \( G^* = (V, E^*) \). Such a 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 \( u \) to \( v \) in \( G \) if and only if there is the edge \( (u, v) \) in \( E^* \).
Definition 2.1. Let $\Gamma$ be a family of graphs corresponding to partially ordered hierarchies. A hierarchical key assignment scheme for $\Gamma$ is a pair $(\text{Gen}, \text{Der})$ of algorithms satisfying the following conditions:

1. The information generation algorithm $\text{Gen}$ is probabilistic polynomial time. It takes as inputs the security parameter $1^\ell$ and a graph $G = (V, E)$ in $\Gamma$, and produces as outputs
   (a) a private information $s_u$ and a key $k_u$, for any class $u \in V$;
   (b) a public information $pub$.
   We denote by $(s, k, pub)$ the output of the algorithm $\text{Gen}(1^\ell, G)$, where $s$ and $k$ denote the sequences of private information and of keys, respectively.

2. The key derivation algorithm $\text{Der}$ is deterministic polynomial-time. It takes as inputs the security parameter $1^\ell$, a graph $G = (V, E)$ in $\Gamma$, two classes $u \in V$ and $v \in A_u$, the private information $s_u$ assigned to class $u$ and the public information $pub$, and outputs the key $k_v$ assigned to class $v$.

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 $\text{Gen}$ on inputs $1^\ell$ and $G$, it holds that $\text{Der}(1^\ell, G, u, v, s_u, pub) = k_v$.

In order to evaluate the security of the scheme, we consider a static adversary which wants to attack a class $v \in V$ and which is able to corrupt all users not allowed to compute the key $k_v$. We define an algorithm $\text{Corrupt}$, which, on input the private information $s$ generated by the algorithm $\text{Gen}$, extracts the secret values $s_u$ associated to all classes $u$ in the set of nodes that do not have access to node $v$, i.e., $F_v = \{ u \in V : v \not\in A_u \}$. We denote by $\text{corr}_v$ the sequence output by $\text{Corrupt}(s)$.

If $A(\cdot, \cdot, \ldots)$ is any probabilistic algorithm then we denote by $a \leftarrow A(x, y, \ldots)$ the experiment of running $A$ on inputs $x, y, \ldots$ and letting $a$ be the outcome. Similarly, if $X$ is a set then $x \leftarrow X$ denotes the experiment of selecting an element uniformly at random from $X$ and assigning $x$ this value. A function $\epsilon : N \to R$ is negligible if for every constant $c > 0$ there exists an integer $r_\epsilon$ such that $\epsilon(\ell) < \ell^{-c}$ for all $\ell \geq r_\epsilon$.

We consider two different security goals: against key recovery and with respect to key indistinguishability. In the key recovery case, the adversary, on input all public information generated by the algorithm $\text{Gen}$, as well as the private information $s_u$, held by corrupted users, outputs a string $k_v^*$ and succeeds whether $k_v^* = k_v$. We require that the adversary will succeed with probability only negligibly different from $1/(2^{\text{length}(k_v)})$.

Definition 2.2 (REC-ST). Let $\Gamma$ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E) \in \Gamma$ be a graph and let $(\text{Gen}, \text{Der})$ be a hierarchical key assignment scheme for $\Gamma$. Let $\text{STAT}_v$ be a static adversary which attacks a class $v$. Consider the following experiment:

\begin{align*}
\text{Experiment } & \text{Exp}_{\text{STAT}_v}^{\text{REC}}(1^\ell, G) \\
& (s, k, pub) \leftarrow \text{Gen}(1^\ell, G) \\
& \text{corr}_v \leftarrow \text{Corrupt}_v(s) \\
& k_v^* \leftarrow \text{STAT}_v^{\text{REC}}(1^\ell, G, pub, \text{corr}_v) \\
& \text{return } k_v^*
\end{align*}

The advantage of $\text{STAT}_v$ is defined as $\text{Adv}_{\text{STAT}_v}^{\text{REC}}(1^\ell, G) = \Pr[k_v^* = k_v]$. The scheme is said to be secure in the sense of REC-ST if, for each graph $G = (V, E) \in \Gamma$ and each class $v \in V$, the function $\text{Adv}_{\text{STAT}_v}^{\text{REC}}(1^\ell, G)$ is negligible for each static adversary $\text{STAT}_v$ whose time complexity is polynomial in $\ell$.

In the key indistinguishability case, two experiments are considered. In the first one, the adversary is given as a challenge the key $k_v$, whereas, in the second one, it is given a random string $\rho$ having the same length as $k_v$. It is the adversary’s job to determine whether the received challenge corresponds to $k_v$ or to a random string. We require that the adversary will succeed with probability only negligibly different from $1/2$.

Definition 2.3 (IND-ST). Let $\Gamma$ be a family of graphs corresponding to partially ordered hierarchies, let $G = (V, E) \in \Gamma$ be a graph in $\Gamma$, let $(\text{Gen}, \text{Der})$ be a hierarchical key assignment scheme for $\Gamma$ and let $\text{STAT}_v$ be a static adversary which attacks a class $v$. Consider the following two experiments:

\begin{align*}
\text{Experiment } & \text{Exp}_{\text{STAT}_v}^{\text{IND}}(1^\ell, G) \\
& (s, k, pub) \leftarrow \text{Gen}(1^\ell, G) \\
& \text{corr}_v \leftarrow \text{Corrupt}_v(s) \\
& d \leftarrow \text{STAT}_v^{\text{IND}}(1^\ell, G, pub, \text{corr}_v, k_v) \\
& \text{return } d
\end{align*}

The advantage of $\text{STAT}_v$ is defined as

\begin{align*}
\text{Adv}_{\text{STAT}_v}^{\text{IND}}(1^\ell, G) &= |\Pr[\text{Exp}_{\text{STAT}_v}^{\text{IND}}(1^\ell, G) = 1] - \Pr[\text{Exp}_{\text{STAT}_v}^{\text{IND}}(1^\ell, G) = 1]|.
\end{align*}

The scheme is said to be secure in the sense of IND-ST if, for each graph $G = (V, E) \in \Gamma$ and each $v \in V$, the function $\text{Adv}_{\text{STAT}_v}^{\text{IND}}(1^\ell, G)$ is negligible, for each static adversary $\text{STAT}_v$ whose time complexity is polynomial in $\ell$.

In Definitions 2.2 and 2.3 we have considered a static adversary attacking a class. A different kind of adversary, the adaptive one, could also be considered. Such an adversary is first allowed to access all public information as well as all private information of a number of users; afterwards, it chooses the class $u$ against which the attack will be mounted. In [4] it has been proven that security against adaptive adversaries is (polynomially) equivalent to security against static adversaries. Hence, in this paper we will only consider static adversaries.
3. Complexity assumptions

An RSA generator with associated security parameter \( \tau \) is a randomized algorithm that returns a pair \((n, e)\), \((n, p, q, d)\), where \( n \) is the RSA modulus, \( e \) is the encryption exponent and \( d \) is the decryption exponent, satisfying the following conditions:

- \( p \) and \( q \) are two distinct large odd primes of \( \tau \) bits;
- \( n = p \cdot q \);
- \( e \in \mathbb{Z}_\phi(n)^* \), where \( \phi(n) = (p - 1) \cdot (q - 1) \);
- \( d = e^{-1} \mod \phi(n) \).

Two strategies to compute the pair \((n, e)\), \((n, p, q, d)\) are used. The former, first chooses the primes \( p \) and \( q \), computes \( n \), picks \( e \) at random in \( \mathbb{Z}_\phi(n)^* \), and computes \( d \) accordingly. Such a strategy yields a random exponent RSA generator, denoted \( K_{\text{RSA}}^{\text{ran}}(1^\tau) \). The latter, fixes the encryption exponent \( e \) to be a small odd number, like 3, 17, or \( 2^{16} + 1 \), and then generates the other parameters, accordingly.\(^1\) Given a fixed odd number \( e \), such a strategy yields an RSA generator for exponent \( e \), denoted \( K_{\text{RSA}}^{\text{fix}}(1^\tau, e) \). The random exponent RSA generators \( K_{\text{RSA}}^{\text{ran}}(1^\tau) \) and \( K_{\text{RSA}}^{\text{fix}}(1^\tau, e) \), are described as follows:

Let \( B \) and Grsa be algorithms where the algorithm Grsa corresponds either to \( K_{\text{RSA}}^{\text{ran}}(1^\tau) \) or to \( K_{\text{RSA}}^{\text{fix}}(1^\tau, e) \). Consider the following experiment:

\[
\text{Experiment } \text{Exp}_B^{\text{Grsa}}(n, e, p, q, d) \leftarrow \text{Grsa}
\]

- \( x \leftarrow \mathbb{Z}_n^* \)
- \( y \leftarrow x^e \mod n \)
- \( x' \leftarrow B(n, e, y) \)
- \( \text{if } x' = x \text{ then return } 1 \)
- \( \text{else return } 0 \)

The advantage of \( B \) is defined as \( \text{Adv}_B^{\text{Grsa}} = \Pr[\text{Exp}_B^{\text{Grsa}} = 1] \).

The RSA generators described above yield the following two assumptions.\(^2\)

**Random Exponent RSA Assumption.** The function \( \text{Adv}_B^{\text{ran}}(1^\tau) \) is negligible, for each probabilistic algorithm \( B \) whose time complexity is polynomial in \( \tau \).

**RSA Assumption for Exponent in a set of odd numbers.** Let \( X \) be a set of odd numbers. For each \( e \in X \), the function \( \text{Adv}_B^{\text{RSA}}(1^\tau, e) \) is negligible, for each probabilistic algorithm \( B \) with time complexity polynomial in \( \tau \).

4. The Akl–Taylor scheme

In this section we describe the Akl and Taylor scheme [1].

Let \( \Gamma \) be a family of graphs corresponding to partially ordered hierarchies, and let \( G = (V, E) \in \Gamma \).

**Algorithm Gen(1\(^\tau\), G):**

1. Randomly choose two distinct large primes \( p \) and \( q \) having bitlength \( \tau \) and compute \( n = p \cdot q \);
2. For each class \( v \in V \), compute an integer \( t_v \), such that \( t_v \) divides \( t_e \) if and only if \( v \in A_v \) (more details are provided immediately after the description of the algorithms);
3. Let \( \text{pub} \) be the sequence of public information computed in the previous step, along with the value \( n \);
4. Randomly choose a secret value \( k_0 \), where \( 1 < k_0 < n \);
5. For each class \( v \in V \), compute the private information \( s_v \) and the encryption key \( k_v \), as follows:
   - \( s_v = k_v = k_0^v \mod n \);
6. Let \( s \) and \( k \) be the sequences of private information and keys, respectively, computed in the previous step;
7. Output \((s, k, \text{pub})\).

\(^1\) There have been some questions raised about the security of this strategy, since it might be possible that roots of small degree are easier to take than roots of a random degree.

\(^2\) To the best of our knowledge, in the literature there is no analysis of the relationships between the security of these two different strategies.
Algorithm Der(1^t, G, u, v, s_u, pub)

Extract the values t_v and t_e from pub and compute
\[ s_{u,v}^{\ell/v} \mod n = (k_{v,u}^{\ell/v}) \mod n = k_v. \]

A crucial issue is how to perform step 2. of Algorithm Gen(1^t, G). Akli and Taylor [1] and MacKinnon et al. [17] proposed two different public values assignments such that for each pair of public values \( t_v \) and \( t_e \), the value \( t_v \) divides \( t_e \) if and only if \( v \in A_u \).

The Akl–Taylor assignment. For each \( v \in V \), choose a distinct prime number \( p_v \) and compute the public value \( t_v \) as follows:
\[
\begin{aligned}
t_v &= \begin{cases} 
1 & \text{if } A_v = V; \\
\prod_{u \in A_v} p_u & \text{otherwise.}
\end{cases}
\end{aligned}
\]

The MacKinnon et al. assignment. Decompose the partially ordered hierarchy \( G = (V, E) \) into disjoint chains (a chain is a totally ordered set) and assign each chain a distinct prime number. Then, for each \( v \in V \), compute the prime power \( n_v = p^v \), where \( v \) is the \( i \)th class in the chain whose assigned prime is \( p \). Finally, for each class \( v \in V \), compute the public value \( t_v \) as follows:
\[
\begin{aligned}
t_v &= \begin{cases} 
1 & \text{if } A_v = V; \\
\lcm_{u \neq v} n_u & \text{otherwise.}
\end{cases}
\end{aligned}
\]

The left-hand side of Fig. 1 shows a partially ordered hierarchy, whereas, one of its chain decomposition is shown the right-hand side.

Example of the MacKinnon assignment.

Consider the hierarchy on the left-hand side of Fig. 1, when the primes assigned to the classes are \( p_a = 3 \), \( p_b = 5 \), \( p_c = 7 \), \( p_d = 11 \), \( p_e = 13 \). The public values are \( t_a = 5 \), \( t_b = 3 \cdot 7 \), \( t_c = 3 \cdot 5 \cdot 7 \), \( t_d = 3 \cdot 5 \cdot 7 \cdot 11 \), \( t_e = 3 \cdot 5 \cdot 7 \cdot 11 \).

Example of the Akl–Taylor assignment.

Consider the chain decomposition shown in Fig. 1, when the prime powers assigned to the classes are \( n_a = 3 \), \( n_b = 5 \), \( n_c = 7^2 \), \( n_d = 7^2 \cdot 11 \), \( n_e = 7^2 \cdot 11^2 \). The public values are \( t_a = 5 \), \( t_b = 3^2 \), \( t_c = 3 \cdot 5 \cdot 7 \cdot 11 \), \( t_d = 3^2 \cdot 5 \), \( t_e = 3^2 \cdot 5^2 \).

Akl and Taylor noticed that in order to construct an Akl–Taylor scheme which is resistant to collusive attacks it is needed that for each \( v \in V \) and each \( X \subseteq F_{j} \), \( \gcd(t_{u} : u \in X) \) does not divide \( t_{e,v} \). Indeed, they showed the following result:

Lemma 4.1 [11]. Let \( t \) and \( t_1, \ldots, t_m \) be integers, and let \( k \in \mathbb{Z}_{n} \), where \( n = p \cdot q \) is the product of two large primes. The power \( k^t \mod n \) can be feasibly computed from the set of powers \( \{ k^{t_1} \mod n, \ldots, k^{t_m} \mod n \} \) if and only if \( \gcd(t_{1}, \ldots, t_{m}) \) divides \( t \).

The proof of Lemma 4.1 relies on the infeasibility of extracting \( t \)th roots modulo \( n \), where \( t > 1 \) is an integer and \( n \) is the product of two large unknown primes. Lemma 4.1 gives an intuition for the security of the scheme but it has never been shown whether the existence of an efficient adversary breaking the security of the scheme in the sense of REC–ST implies the existence of an adversary which efficiently solves a computational hard problem.

4.1. Proving the security of the Akl–Taylor scheme

In this section we show that the Akl–Taylor scheme, according to both Akl–Taylor and MacKinnon et al. assignments, is secure against key recovery provided that the primes associated to the classes are properly chosen.

In the following we describe two primes choices. The first choice, denoted fixed primes choice, yields instances of the Akl–Taylor scheme secure under the RSA assumption for exponent in a set of odd numbers. The second one, denoted \( R \)-random primes choice, yields instances secure under the random exponent RSA assumption.

- **Fixed primes choice.** Let \( \text{PRIMES}_f = \{ p_1, \ldots, p_\ell \} \) be the set of the first \( \ell \) prime numbers greater than two.
  - Let \( u_1, \ldots, u_{|V|} \) be a sorting of \( V \). In the Akl–Taylor assignment, associate prime \( p_j \in \text{PRIMES}_f \) to class \( u_j \), for each \( j = 1, \ldots, \ell \).
  - Let \( m \) be the number of disjoint chains in the minimal decomposition of \( G \) and let \( u_j \) be the first node of the \( j \)th chain. In the MacKinnon et al. assignment, associate prime \( p_j \in \text{PRIMES}_m \) to class \( u_j \), for each \( j = 1, \ldots, m \).
- **\( R \)-Random primes choice.** Let \( R = \{ R_n \} \) be a family of sets of integers, where \( n \) is an RSA modulus. In the Akl–Taylor assignment (in the MacKinnon et al. assignment, resp.), choose uniformly at random, for each \( n \) (for each chain in the minimal chain decomposition of the partially ordered hierarchy, resp.), a distinct prime number in \( R_n \), for an appropriately chosen family \( R \).

\[ \text{Let } \text{Gen}(s, K, \text{PRIMES}_s) \text{ be a family of algorithms, where } \text{PRIMES}_s \text{ is the set of the first } \ell \text{ prime numbers greater than two.} \]

\[ \text{Let } u_1, \ldots, u_{|V|} \text{ be a sorting of } V \text{. In the Akl–Taylor assignment, associate prime } p_j \in \text{PRIMES}_f \text{ to class } u_j \text{, for each } j = 1, \ldots, \ell \text{.} \]

\[ \text{Let } m \text{ be the number of disjoint chains in the minimal decomposition of } G \text{ and let } u_j \text{ be the first node of the } j \text{th chain. In the MacKinnon et al. assignment, associate prime } p_j \in \text{PRIMES}_m \text{ to class } u_j \text{, for each } j = 1, \ldots, m \text{.} \]

\[ \text{Let } R = \{ R_n \} \text{ be a family of sets of integers, where } n \text{ is an RSA modulus. In the Akl–Taylor assignment (in the MacKinnon et al. assignment, resp.), choose uniformly at random, for each } n \text{ (for each chain in the minimal chain decomposition of the partially ordered hierarchy, resp.), a distinct prime number in } R_n \text{, for an appropriately chosen family } R \text{.} \]

---

3 The problem of finding a decomposition into a minimum number of chains is solvable in time polynomial in \(|V|\) (for example by using Khachiyan’s algorithm [15]).

4 In particular, when \( \gcd(r, \phi(n)) = 1 \), where \( \phi(n) \) is the Euler’s totient function, this is the assumption behind the RSA cryptosystem [19]; whereas, if \( r = 2 \), this assumption is used in the Rabin cryptosystem [18].
The Akl–Taylor assignment. We start by considering the fixed primes choice for the Akl–Taylor assignment. It holds that:

**Theorem 4.2.** Let $G = (V, E)$ be a partially ordered hierarchy. The Akl–Taylor assignment with the fixed primes choice yields a scheme which is secure in the sense of REC–ST under the RSA assumption for exponents in PRIMES$_{V}$. 

**Proof.** We show how to turn a polynomial-time static adversary breaking the security of the Akl–Taylor scheme in the sense of REC–ST, when the public parameters are chosen according to the fixed primes choice, into a polynomial-time adversary $B$ breaking the RSA assumption for exponents in PRIMES$_{V}$.  

Let $e \in$ PRIMES$_{V}$ and assume, w.l.o.g., that $e = p$. Let $u_{1}, \ldots, u_{|V|}$ be a sorting of the classes in $V$. Assume that there exists a static adversary $\text{STAT}_{u_{i}}$ which is able to compute, with non-negligible advantage, the key $k_{u_{i}}$ held by the class $u_{i}$ in the Akl–Taylor scheme. We construct a polynomial-time adversary $B$ that, on input a triple $(n, e, y)$, where $n$ is chosen according to the RSA generator $\mathcal{L}_{\text{RSA}}(1^{|e|}, e)$, uses the adversary $\text{STAT}_{u_{i}}$ to compute, with non-negligible advantage, a value $x \in \mathbb{Z}_{n}^{*}$ such that $y = x^{e} \mod n$. The adversary $B$ is described in the following.

**Algorithm B(n, e, y)**

1. Construct the inputs for $\text{STAT}_{u_{i}}$ as follows:
   (a) For each $j = 1, \ldots, |V|$, assign prime $p_{j}$ to class $u_{j}$ (notice that $p_{j} = e$ is assigned to $u_{j}$);
   (b) For each $j = 1, \ldots, |V|$, compute the integer $t_{u_{j}} = \prod_{i \notin A_{u_{j}}} p_{z}$ ($t_{u_{j}} = 1$ if $A_{u_{j}} = V$);
   (c) Set $pub$ to be the sequence of public information computed in the previous step, along with the value $n$;
   (d) For each class $v \in F_{u_{i}}$, compute the key $k_{v} = y^{v^{e}/p_{v}} \mod n$;
   (e) Set $corr_{u_{i}}$ to be the sequence of keys computed in the previous step;
2. Let $k_{u_{i}}$ be the output of $\text{STAT}_{u_{i}}$ on input $(1^{e}, G, pub, corr_{u_{i}})$;
3. Use Extended Euclidean Algorithm to compute integers $\alpha$ and $\beta$ such that $p_{i} \cdot \alpha + t_{u_{i}} \cdot \beta = \gcd(p_{i}, t_{u_{i}}) = 1$ (see [20]);
4. Compute $x = y^{\alpha} \cdot k_{u_{i}}^{\beta} \mod n = x^{\phi(n)} \mod n \cdot x^{\phi(n) \beta} \mod n = x^{\phi(n) \alpha + t_{u_{i}} \beta} \mod n$;
5. Output $x$.

Notice that, in step 1.(d), the key $k_{v}$ can be computed, since $p_{i}$ divides $t_{v}$. Moreover, $\text{STAT}_{u_{i}}$’s view in the above simulation is identically distributed as the one obtained in a real execution of the scheme with the fixed primes choice, and all the computations needed to construct such a view can be performed in polynomial time. Finally, it is easy to see that $\text{Adv}_{\text{Euler}}(1^{e}, e) = \text{Adv}_{\text{RSA}}(1^{e})$. Since $\text{Adv}_{\text{RSA}}(1^{e})$ is non-negligible, the theorem holds.

Let us consider the R-random primes choice for the Akl–Taylor assignment. The next result establishes a lower bound for the Euler totient function $\phi(n)$, when $n$ is an RSA modulus obtained as the product of two large distinct primes having bitlength $\tau$. More precisely:

**Lemma 4.3.** Let $p$ and $q$ two large distinct primes having bitlength $\tau$ and let $n = p \cdot q$. The Euler totient function $\phi(n)$ satisfies

$$\phi(n) > 2^{2^{\tau-2}} - 2^{\tau}.$$ 

**Proof.** Let $p$ and $q$ two large primes having bitlength $\tau$. Thus, $p, q > 2^{\tau-1}$. Therefore,

$$\phi(n) = (p - 1) \cdot (q - 1) = p \cdot q - p - q + 1$$
$$> 2^{2^{\tau-2}} - 2 \cdot 2^{\tau-1} = 2^{2^{\tau-2}} - 2^{\tau}. \quad \blacksquare$$

We fix the family $R$ in the R-random primes choice by setting $R_{n}$ equal to the set of integers belonging to the interval $[3, w]$, where $w = 2^{2^{\tau-2}} - 2^{\tau}$. Therefore, to avoid overburdening the notation, in the following we will refer to the R-random primes choice simply as to the random primes choice.

The following result holds:

**Theorem 4.4.** Let $G = (V, E)$ be a partially ordered hierarchy. The Akl–Taylor assignment with the random primes choice yields a scheme secure in the sense of REC–ST under the random exponent RSA assumption.

In order to prove Theorem 4.4 we need some results from Number Theory, which can be found in [13]. Let $x$ be a positive integer and let $\pi(x)$ be the prime counting function, i.e., the number of prime numbers less than or equal to $x$. The Prime Number Theorem states that $\pi(x)$ has asymptotic behavior $\pi(x) \sim x / \ln x$, where $\ln x$ denotes the natural logarithm of $x$. In particular, for any $x \geq 17$ it holds that

$$\pi(x) > x / \ln x.$$ 

(1)

The number of different prime factors of an integer $x$ is denoted by $\omega(x)$ and has normal order $\log \log x$ (see [13, pag. 356]). In particular, Erdős and Pomerance [9] proved that the number of prime factors of $\phi(n)$ has normal order $(\log \log n)^2 / 2$, that is,

$$\omega(\phi(n)) \sim (1 - \epsilon) (\log \log n)^2 / 2 < \omega(\phi(n)) < (1 + \epsilon) (\log \log n)^2 / 2,$$

for each $\epsilon > 0$ and almost all values of $n$. This means that there might be an exceptional infinitesimal set of values, i.e., $o(n)$ values, for which the above inequalities are false, and this exceptional set will naturally depend upon $\epsilon$.

The next result shows that the probability of choosing a prime number $p_{e} \in [3, w]$ which is not relatively prime to $\phi(n)$ is negligible.

**Lemma 4.5.** Let $w = 2^{2^{\tau-2}} - 2^{\tau}$, and let $P_{w} = \{a < w : a$ is prime$\}$. For almost all RSA modulus $n$ having bitlength at most $2\tau$, the fraction $|P_{w} - Q_{\omega}| / |P_{w}|$, where $Q_{\omega} = \{a < w : a$ is prime and $\gcd(a, \phi(n)) = 1\}$, is a negligible function in $\tau$. 

---

Proof. Notice that $|P_w| = \pi(w)$, i.e., the number of prime numbers less than or equal to $w$, and $|Q_w| \geq |P_w| - \omega(\phi(n))$, with equality when all prime factors of $\phi(n)$ belong to $P_w$. Since for any $\tau > 3, w > 2^{2\tau - 3}$ and $w, n < 2^{2\tau}$, it follows that

$$\frac{|P_w| - |Q_w|}{|P_w|} \leq \frac{\omega(\phi(n))}{\pi(w)} \leq \frac{\omega(\phi(n))}{w} \leq \frac{1}{\tau} \cdot \frac{\omega(\phi(n))}{w} \leq \frac{1}{\tau} \cdot \ln w \cdot w \quad (from \ (1) \ and \ (2))$$

We are now ready for the proof of Theorem 4.4.

Assume there exists a static adversary $\text{STAT}_n$ which is able to compute, with non-negligible advantage, the key $k_n$ held by a class $u \in V$ in the Akl–Taylor scheme, when the public parameters are chosen according to the random primes choice. We construct a polynomial-time adversary $B$ that, on input a triple $(n, e, y)$, where $n$ and $e$ are chosen according to the random exponent RSA generator $K_{\text{gen}}^{\text{RSA}}(1^*), \text{uses}$ the adversary $\text{STAT}_n$ to compute, with non-negligible advantage, a value $x \in \mathbb{Z}_n^*$ such that $y = x^e \mod n$. Let $\text{PRIMES}$ denote the set of prime numbers. The adversary $B$ is described in the following.

Algorithm $B(n, e, y)$

1. If $e \not\in [3, w] \cap \text{PRIMES}$ then output ‘failed’
2. Set $p_u = e$ and for each class $v \in V \setminus \{u\}$, choose uniformly at random a distinct prime number $p_v$ in $[3, w] \cap (\text{PRIMES} \setminus \{e\})$.
3. For each class $v \in V$, compute the integer $t_v = \prod_{p_i \in \text{PRIMES}} p_i (t_v = 1 \text{ if } A_v = V)$.
4. Set $p_b$ to be the sequence of public information computed in the previous step, along with the value $n$;
5. For each class $v \in F_u$, compute the key $k_v = y^{\frac{t_v}{n}} \mod n$;
6. Set $\text{corr}_n$ to be the sequence of keys computed in the previous step;
7. Let $k_n$ be the output of $\text{STAT}_n$ on input $(1^*, L, \text{pub}, \text{corr}_n)$;
8. Use Extended Euclidean Algorithm to compute integers $\alpha$ and $\beta$ such that $p_u \cdot \alpha + t_u \cdot \beta = \gcd(p_u, t_u) = 1$ (see [20]);
9. Compute $x = y^\beta \mod n$;
10. Output $x$.

Notice that, in step 4., the key $k_v$ can be computed, since $p_v$ divides $t_v$.

Let $\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$ denote the advantage of the adversary $\text{STAT}_n$ of breaking the Akl–Taylor scheme when the primes assigned to the classes are chosen according to the Akl–Taylor assignment with random primes choice and let $\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$ be the advantage over the choices where $p_u$ satisfies $\gcd(p_u, \phi(n)) = 1$. Due to Lemma 4.5, if $\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$ is non-negligible, then $\text{STAT}_n$ breaks the scheme when the prime $p_u$ is such that $\gcd(p_u, \phi(n)) = 1$. Indeed, if $\text{STAT}_n$ were able to break only on assignments where $p_u$ satisfies $\gcd(p_u, \phi(n)) \neq 1$, then $\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$ would have been overall negligible. Hence, “since $w = 2^{2\tau - 2} - 2^\tau$ and $\phi(n), w, n < 2^{2\tau}$, for any constant $c > 2$ and sufficiently large $\tau$ it holds that:

$$\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*) = \text{Prob}(e \leq w \text{ is a prime and } \gcd(e, \phi(n)) = 1) \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$$

$$\geq \frac{\pi(w) - \omega(\phi(n))}{\pi(w)} \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*) \quad (\text{from } (1))$$

$$\geq \frac{w - \ln w \cdot \omega(\phi(n))}{\ln w \cdot \phi(n)} \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$$

$$\geq \frac{2^{2\tau - 2} - 2^\tau - 2^\tau - \omega(\phi(n))}{2^\tau - 2^\tau} \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$$

$$\geq \frac{2^\tau - 2^\tau - (1 + \epsilon) \cdot (\ln \ln n)^2}{2^\tau - 2^\tau} \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*) \quad (\text{from } (2))$$

$$\geq \frac{1}{c \cdot \tau} \cdot \text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*) \quad (\text{from } (3))$$

Since $\text{Adv}_{\text{STAT}_n}^\text{KAR}(1^*)$ is non-negligible also $\text{Adv}_{\text{KAR}}^\text{KAR}(1^*)$ is non-negligible. This concludes the proof.

The Mackinnon et al. assignment. Consider the fixed primes choice for the Mackinnon et al. assignment. Next result holds:

Theorem 4.6. Let $\mathbb{G} = (V, E)$ be a partially ordered hierarchy, and let $m$ be the number of chains of a decomposition of $\mathbb{G}$. The Mackinnon et al. assignment with the fixed primes choice in $\text{PRIMES}_m$ yields a scheme which is secure in the sense of REC–ST under the RSA assumption for exponents in $\text{PRIMES}_m$. 


Theorem 4.7. The Akl–Taylor assignment with the modified random primes choice yields a scheme which is secure in the sense of REC–ST under the random exponent RSA assumption.

Proof. Let \( B \) be an adversary defined as follows:

Algorithm \( B(n, e, y) \)

1. If \( e \notin [3, w] \cap \text{PRIMES} \) then output 'failed'
2. For each \( j = 1, \ldots, m \), assign \( p_j \) to class \( u_j \) (notice that \( p_1 = e \) is assigned to \( u_1 \));
3. For each \( j = 1, \ldots, |V| - i \), assign prime \( p_{j+i} \) to class \( u_j \);
4. For each class \( v \in V \), compute the public value \( t_v \) as follows:
   \[
   t_v = \begin{cases} 1 & \text{if } A_u = V; \\ \text{lcm}_{u_j \in \text{class}} n_{u_j} & \text{otherwise}. \end{cases}
   \]
5. Set \( \text{pub} \) to be the sequence of public information computed in the previous step, along with the value \( n \);
6. Compute \( x = y^o \mod n \cdot \ell \mod n = y^o \mod n \cdot (y^{\frac{1}{d}})^{\ell} \mod n = x^{\frac{1}{d}} \mod n \).
7. Output \( x \).

Notice that, in step 1(e), the key \( k_i \) can be computed, since \( e^i \) divides \( t_v \). Moreover, notice that \( \gcd(e, t_{j+i}/e^{i-1}) = 1 \), since \( e^{i-1} \) is the public power associated to the class which precedes \( u \) in the chain, to whom \( u \) has not access to. Thus, each class can view the sequence of public information in the previous step, along with the value \( n \).

With the modified random primes choice, only the first prime \( n \) needs to be published, along with the modulus \( n \). Indeed, such public information allows each class to compute the other primes and the sequence of integers \( t_v 's needed for key derivation. These computations can be performed efficiently (see [2] where a similar technique was employed). Notice that, in the modified random primes choice, \( i \geq 2 \) primes of the sequence \( p_1, \ldots, p_{|V|} \) might be published. Then, each class \( v \) has to compute \( |V| - i \) missing primes accordingly, by performing on average \( O(|V| - i \cdot \tau) \) steps to compute the full sequence. This allows to obtain a tradeoff between the size of the public information and the number of steps required to perform key derivation.
5. Enforcing more general access control policies

Due to its simplicity and versatility, the Akl–Taylor scheme has been used as a starting point in the design of key assignment schemes for the enforcement of more general access control policies, including those with transitive and anti-symmetrical exceptions [7,22], as well as those with time-dependent constraints [21,23]. As shown in [4,7], such policies can be represented by a directed acyclic graph \( G = (V,E) \) with \( V = R \cup N \), where \( R \) and \( N \) denote the set of root classes and non-root classes, respectively. In particular, the classes in \( R \) do not need to be assigned encryption keys, since they have no data to be protected.

Given a graph \( G = (V,E) \), with \( V = R \cup N \), consider the following assignment, which we refer to as the reduced Akl–Taylor assignment:

1. For each class \( u \in R \), let \( p_u = 1 \);
2. For each class \( v \in N \), choose a distinct prime number \( p_v \);
3. For each class \( u \in V \), compute the public value \( t_u \) as done in the Akl–Taylor assignment.

The reduced assignment satisfies the property that, for each \( v \in N \), \( \gcd(t_u: u \in F_v) \) does not divide \( t_v \) (see [7]). Moreover, the assignment is similar to the Akl–Taylor one, but needs a smaller amount of public information, i.e. \( |N| \) primes instead of \( |V| \), since primes are assigned only to non-root classes (this is the reason why the assignment is said to be reduced).

Examples of Akl–Taylor based schemes using the reduced assignment can be found in [7,21,23]. More precisely, the constructions in [7,21] first transform the graph representing the access control policy into a bipartite graph corresponding to a two-level partially ordered hierarchy, where the classes at the first level are root classes, whereas, those at the second level correspond to non-root classes. Then, the Akl–Taylor scheme with the reduced assignment is used on such a hierarchy. The construction in [23] is essentially the same as the one proposed in [21], but uses a different graph transformation.

The next remark will be useful when analyzing the security of the reduced assignment.

Remark 5.1. Let \( G = (V,E) \), with \( V = R \cup N \), be a graph corresponding to a partially ordered hierarchy and let \( p_v \) denote the prime associated to class \( v \in V \) by the Akl–Taylor assignment. Consider two instances of the Akl–Taylor scheme using the Akl–Taylor assignment and the Akl–Taylor reduced assignment where each class \( v \in N \) is assigned the prime \( p_v \), respectively. Let \( k_0 \) (\( k'_0 \), resp.) denote the secret value chosen in step 4 of the key generation algorithm in the first (second, resp.) instance of the scheme. Moreover, let \( k_v \) (\( k'_v \), resp.) denote the key assigned to class \( v \) by the first (second, resp.) instance of the scheme.

It is easy to see that if \( k_0 = (k_0)^{\prod_{v \in R} p_v} \mod n \), then

\[ k'_v = (k'_0)^{p_v} \mod n, \quad \text{for any } v \in R; \]
\[ k_v = k'_v, \quad \text{for any } v \in N. \]

For example, consider the hierarchy of Fig. 1, where \( R = \{a, b\} \). The keys assigned to the classes by the first instance are \( k'_a = (k'_0)^{p_a} \mod n, k'_b = (k'_0)^{p_b} \mod n, k'_c = (k'_0)^{p_c} \mod n, k'_d = (k'_0)^{p_d} \mod n, k'_e = (k'_0)^{p_e} \mod n, k'_f = (k'_0)^{p_f} \mod n \), whereas, those assigned by the second instance are \( k''_a = k''_b = (k''_0)^{p_a} \mod n, k''_c = (k''_0)^{p_c} \mod n \). The next result holds:

Theorem 5.2. The reduced Akl–Taylor assignment with the fixed primes choice yields an Akl–Taylor scheme which is secure in the sense of REC–ST, with respect to classes in \( N \), under the RSA assumption for exponents in PRIMES\(_N\).

Proof. We show how to turn a polynomial-time static adversary breaking the security of the Akl–Taylor scheme in the sense of REC–ST, when the public parameters are chosen according to the reduced Akl–Taylor assignment with fixed primes choice, into a polynomial-time static adversary breaking the security of the Akl–Taylor scheme in the sense of REC–ST, when the public parameters are chosen according to the Akl–Taylor scheme with the fixed primes choice. Then, the theorem follows from Theorem 4.2.

Assume that there exists a static adversary \( \text{STAT}_a \) which is able to compute, with non-negligible advantage, the key \( k''_u \) held by a non-root class in \( G'' = (V'', E'') \) in the Akl–Taylor scheme when the reduced assignment with the fixed primes choice is used. We construct an adversary \( \text{STAT}'_a(1^\tau, G', pr', corr') \), where \( G' = (V', E') \) and \( pr' \) denote the sequence of primes assigned to the classes in \( V' \) (recall that the sequence \( pub' \) of integers can be easily computed from \( pr' \), with at most \( O(|V'|^2) \) operations), as follows:

Algorithm \( \text{STAT}'_a(1^\tau, G', pr', corr') \):

1. Construct an input instance for \( \text{STAT}'_a(1^\tau, G', pr', corr') \) in which implicitly the secret value \( k''_0 \) is set to \( (k''_0)^{\prod_{v \in R'} p_v} \mod n \), where \( R' \) denotes the set of root classes in \( G' \), as follows
   (a) Let \( G' = G'' \);  
   (b) Let \( pr'' \) be the subsequence of \( pr' \) containing the primes assigned to non-root classes in \( V'' \setminus R'' \); 
   (c) Let \( u = v \) be the class to be attacked; 
   (d) Let \( corr'_v \) be the keys held by corrupted users. From Remark 5.1, such keys can be easily constructed from \( corr'_v \); 
2. Let \( k_v' \) be the output of \( \text{STAT}''_u \) on inputs \( (1^\tau, G', pr'', corr'') \); 
3. Output \( k_v' = k''_v \).

It is easy to see that \( \text{STAT}'_a \) has exactly the same advantage of \( \text{STAT}''_a \) in breaking the Akl–Taylor scheme. Therefore, the theorem follows from Theorem 4.2. ■

Along the same lines of the proof of Theorem 5.2, we can show that the Akl–Taylor scheme with the fixed primes choice is secure (with respect to classes in \( N \)) under the random exponent RSA assumption.

We point out that the Akl–Taylor scheme with the reduced assignment allows to obtain key assignment schemes with time-dependent constraints which requires a small amount of public information (see [21,23]). Hence, Theorem 5.2 provides a formal proof for the schemes
in [21,23]. In such schemes each user may be assigned to a class for an arbitrary subset of time periods. The idea is the following: let \( G = (V, E) \) be a graph representing a partially ordered hierarchy and let \( T \) be a sequence of time periods. Consider a two-level hierarchy \( G' = (V', E') \) where \( V' = R \cup N \), \( |R| = |V'| \cdot 2^{|T|} \), and \( |N| = |V'| \cdot |T| \). For each class \( v \in V \) and each subset \( Q \subseteq T \), there is a root class \( u_Q \) in \( R \); whereas, for each \( v \in V \) and each time period \( t \in T \), there is a non-root class \( v_t \in N \). Whenever a user is assigned to a class \( u \in V \) for a subset \( Q \subseteq T \) of time periods, it receives the private information held by class \( u_Q \), which allows to compute the key \( k_u \), for any \( v \in A_u \) and any \( t \in Q \). The amount of public information required by the scheme is \(|V| \cdot |T| \) values, independently of the number of root classes in \( R \).

No other scheme with time-dependent constraints has the same performance. For example, the encryption based construction in [4] requires \( O(|V| \cdot |T|^2) \) public values in the simplest case where each user is assigned to a class for a time-interval and \( O(|V| \cdot 2^{|T|}) \) values if arbitrary subsets of time periods are considered.

6. The Harn–Lin scheme

In 1990, Harn and Lin [14] proposed a key assignment scheme similar to the Akl–Taylor scheme. Their scheme, however, uses a bottom-up key generation process, instead of using a top-down strategy as in the Akl–Taylor scheme. The advantage of their solution is that the size of the public integers \( t_s \)'s assigned to the classes is smaller than that in the Akl–Taylor scheme, yielding a more efficient key derivation for broad and shallow hierarchies. The scheme is described in the following.

Let \( \Gamma \) be a family of graphs corresponding to partially ordered hierarchies and let \( G = (V, E) \in \Gamma \).

**Algorithm Gen(1^*, G)**

1. Randomly choose two distinct large primes \( p \) and \( q \), having bitlength \( r \), and compute \( n = p \cdot q \) and \( \phi(n) = (p - 1)(q - 1) \);
2. For each class \( v \in V \), choose a distinct prime number \( e_v \), such that \( \text{gcd}(e_v, \phi(n)) = 1 \), and compute a public value \( t_v = \prod_{u \in A_v} e_u \);
3. Let \( pub \) be the sequence of public information computed in the previous step, along with the modulus \( n \);
4. For each class \( v \in V \), compute the secret integer \( d_v \), such that \( e_v \cdot d_v = 1 \mod \phi(n) \);
5. Choose a random integer \( k_0 \), such that \( \text{gcd}(k_0, n) = 1 \);
6. For each class \( v \in V \), compute the private information \( s_v \) and the encryption key \( k_v \), as follows:
   \[
   s_v = k_v = \prod_{u \in A_v} d_u \mod n;
   \]
7. Let \( s \) and \( k \) be the sequences of private information and keys, respectively, computed in the previous steps;
8. Output \((s, k, pub)\).

**Algorithm Der(1^*, G, v, u, s_v, pub)**

1. Extract the values \( t_u \) and \( t_v \) from \( pub \);
2. Compute
   \[
   (s_v)^{t_u/\tau_v} \mod n = k_v.
   \]

**Example of Harn–Lin public values.** Consider the hierarchy of Fig. 1, when the primes assigned to the classes are the same as the Akl–Taylor assignment above. The public values associated to the classes are \( t_0 = 3 \cdot 7 \cdot 11 \cdot 13 \), \( t_5 = 5 \cdot 11 \cdot 13 \), \( t_c = 7 \cdot 13 \), \( t_d = 11 \cdot 13 \), \( t_e = 13 \).

Harn and Lin [14] claimed that the security of their scheme is equivalent to that of the RSA cryptosystem. In particular, they analyzed the problem of computing the key \( k_v \) held by a class \( v \in V \) starting from the keys held by a coalition of children of \( v \). However, they did not consider attacks carried out by different kinds of coalitions.

By using a reduction to the Akl–Taylor scheme we prove that, when the public parameters are properly chosen, the Harn–Lin scheme is secure against key recovery. Let \( k_0^AT \) and \( k_0^HL \) be the secret value \( k_0 \) chosen in step 4 of the Akl–Taylor and in step 5 of the Harn–Lin scheme, respectively. The intuition behind our proof is that an instance of the Akl–Taylor scheme yields an instance of the Harn–Lin scheme where \( k_0^HL = (k_0^AT)^{\sum_{v \in V} p_v} \mod n \).

Indeed, with this setting, the key \( k_0^HL \) assigned to \( v \) by the Harn–Lin scheme is equal to that assigned to \( v \) by the Akl–Taylor scheme:

\[
  k_0^HL = (k_0^AT)^{\sum_{v \in V} d_v} \mod n = (k_0^AT)^{\sum_{v \in V} p_v \sum_{u \in A_v} d_u} \mod n = (k_0^AT)^{\sum_{u \in A_v} p_u} \mod n = k_0^AT.
\]

The next result holds:

**Theorem 6.1.** The fixed primes choice in PRIMES, yields an Harn–Lin scheme which is secure in the sense of REC–ST under the RSA assumption for exponents in PRIMES.

**Proof.** We show how to turn a polynomial-time static adversary breaking the security of the Harn–Lin scheme in the sense of REC–ST, when the public parameters are chosen according to the fixed primes choice, into a polynomial-time static adversary breaking the security of the Akl–Taylor scheme in the sense of REC–ST. Thus, the theorem follows from Theorem 4.2.

Assume that there exists a static adversary \( \text{STAT}^*_u \) which is able to compute, with non-negligible advantage, the key \( k_0^HL \) held by a class \( u \in V \) in the Harn–Lin scheme when the fixed primes choice is used. We construct an adversary \( \text{STAT}^*_u(1^*, G, p', corr_u) \), where \( p' \) denote the sequence of primes assigned to the classes in \( V \) (recall that the sequence \( pub' \) of integers can be easily computed from \( p' \), with at most \( O(V' \cdot 2^{|T|}) \) operations), as follows:

**Algorithm STAT^*_u(1^*, G, p', corr_u)**

1. Construct an input instance for \( \text{STAT}^*_u(1^*, G, p, corr_u) \) in which implicitly the secret value \( k_0^ST \) is set to \( k_0^ST \prod_{v \in V} d_v \mod n \), as follows
   \[
   (a) \text{ Let } G = G';
   (b) \text{ Let } p = p';
   (c) \text{ Let } u = v \text{ be the class to be attacked;}
   (d) \text{ Let } corr_u = corr_v \text{ be the keys held by corrupted classes;}
\]
2. Let $k_{\gamma}^{HL}$ be the output of $\text{STAT}_{u}^{HL}$ on inputs $(1^t, G, pr, corr_u)$;
3. Output $k_{\gamma}^{F} = k_{\gamma}^{HL}$.

$\text{STAT}_{u}^{HL}$ has exactly the same advantage in breaking the Akl–Taylor scheme that $\text{STAT}_{u}^{HL}$ has in breaking the Harn–Lin scheme. Therefore, the theorem follows from Theorem 4.2.

Notice that we can consider the random primes choice for the Harn–Lin scheme. Along the same lines of the proof of Theorem 6.1, we can show that the corresponding scheme is secure under the random exponent RSA assumption.

7. Towards security w.r.t. key indistinguishability

In this section we show how to construct a key assignment scheme which is secure with respect to key indistinguishability, starting from a key assignment scheme which guarantees security against key recovery. Our construction uses the Goldreich–Levin hard-core bit (GL bit) [11], since the GL bit is a natural candidate for turning hardness of computation into indistinguishability.

Let $r$ be a security parameter. Given two strings $x = x_1 \cdots x_r$ and $r = r_1 \cdots r_y$ of length $y$ polynomially bounded in $r$, the GL bit $B_r(x)$ corresponds to the inner product (mod 2) of $x$ and $r$, i.e., $B_r(x) = \sum_{i=1}^{y} x_i \cdot r_i \mod 2$. Let $f$ be a one-way permutation and let $\text{Pred}$ be a probabilistic polynomial time algorithm that on input $(f, x)$ and $r$ tries to guess the bit $B_r(x)$. The advantage of $\text{Pred}$ in guessing $B_r(x)$ from $f(x)$ and $r$, where $x$ and $r$ are uniformly chosen in $\{0, 1\}^*$, is defined as

$$\text{Adv}_{\text{Pred}}(1^t) = \left| Pr[\text{Pred}(1^t, f(x), r) = B_r(x)] - \frac{1}{2} \right|.$$

Goldreich and Levin [11] showed that, for every one-way permutation $f$, given $f(x)$ and $r$, it is infeasible to guess $B_r(x)$ with non-negligible advantage. They also showed that, given a polynomial-time algorithm which guesses $B_r(x)$ with non-negligible advantage, it is possible to reconstruct $x$ with non-negligible probability by using the so-called Goldreich–Levin reconstruction algorithm (see [10] for details):

**Theorem 7.1** ([11]). Let $x, r \in \{0, 1\}^{\text{poly}(r)}$. Given an oracle that, on input $r$ and $f(x)$, predicts the value of $B_r(x)$ with advantage $\epsilon(r)$ (over the choice of $r$) in time $t(r)$, there exists a probabilistic polynomial-time algorithm $\text{R}$ with running time $O(\frac{r^{2+\epsilon(r)}}{\epsilon(r)^2})$ that retrieves $x$ with probability at least $\Omega(\epsilon(r))$.

Let $\Gamma$ be a family of graphs corresponding to partially ordered hierarchies, let $\gamma$ be a security parameter and let $\gamma$ be polynomially bounded in $r$. For each graph $G = (V, E)$ in $\Gamma$ we define a graph transformation, whose output, denoted by $G_{\gamma} = (V_{\gamma}, E_{\gamma})$, is called the $\gamma$-extended graph for $G$. We denote by $\Gamma_{\gamma}$ the family of $\gamma$-extended graphs for elements in $\Gamma$. The transformation works as follows:

- For each $u \in V$, we place a class $u_0$ along with $\gamma$ additional classes $u_1, \ldots, u_{\gamma}$, called the classes associated to $u$, in $V_{\gamma}$;
- For each class $u \in V$, we place the edge $(u_0, u_{\gamma+1})$, for $i = 0, \ldots, \gamma - 1$, in $E_{\gamma}$;
- For each $(u, v) \in E$, we place the edge $(u_0, v_0)$ in $E_{\gamma}$.

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

Let $\tau$ be a security parameter and let $\gamma$ be polynomially bounded in $r$. Let $\Gamma'$ be a family of graphs corresponding to partially ordered hierarchies, let $\Gamma_{\gamma}$ be the family of extended graphs for elements in $\Gamma'$, and let $(Gen', Der')$ be a hierarchical key assignment scheme for $\Gamma_{\gamma}$. Now we are ready to describe the proposed key assignment scheme. Let $G \in \Gamma, G_{\gamma} \in \Gamma_{\gamma}$, and let the symbol $\circ$ denote string concatenation.

**Algorithm** $\text{Gen}(1^t, G)$

1. Construct the $\gamma$-extended graph $G_{\gamma} = (V_{\gamma}, E_{\gamma})$ for $G = (V, E)$;
2. Let $(s', k', pub')$ be the output of $\text{Gen'}$ on inputs $(1^t, G_{\gamma})$;
3. Randomly choose a string $r \in \{0, 1\}^{\gamma}$;
4. For each class $u \in V$, let $s_u = s'_{u_0}$;
5. For each class $u \in V$, compute the key $k_u$ as follows:
   $$k_u = B_r(k'_{u_0}) \circ B_r(k'_{u_1}) \circ \cdots \circ B_r(k'_{u_{\gamma-1}});$$
6. Let $s$ and $k$ be the sequences of private information and keys, respectively, computed in the previous steps;
7. Let $pub = (pub', r)$;
8. Output $(s, k, pub)$.

**Algorithm** $\text{Der}(1^t, G, u, v, s_u, pub)$

1. Let $pub' = (pub', r)$;
2. For $i = 1, \ldots, \gamma$, let $k'_{u_i}$ be the output of $\text{Der'}$ on inputs $(1^t, G_{\gamma}, u_0, v_i, s_u, pub')$;
3. Compute $k_u = B_r(k'_{u_0}) \circ B_r(k'_{u_1}) \circ \cdots \circ B_r(k'_{u_{\gamma-1}})$.

Intuitively, since, for each $i = 1, \ldots, \gamma$, it is infeasible to guess $B_r(k_{u_i})$ with non-negligible advantage, an adversary for a class $u \in V$ has no information about even a single bit of the key $k_u$.

The next theorem states that if $(Gen', Der')$ is secure against key recovery, then $(Gen, Der)$ is secure with respect to key indistinguishability.
Theorem 7.2. If (Gen’, Der’) is secure in the sense of REC-ST, then (Gen, Der) is secure in the sense of IND-ST.

Proof. The proof uses a standard hybrid argument. Let $G = (V, E)$ be a graph in $\Gamma'$, let $u \in V$, and let $\text{STAT}_u^{\text{IND}}$ be a static adversary attacking class $u$. In order to prove the theorem, we need to show that the adversary’s views in experiments $\text{Exp}^{\text{IND-0}}_u$ and $\text{Exp}^{\text{IND-1}}_u$ are indistinguishable. We construct a sequence of $\gamma + 1$ experiments $\text{Exp}^0_u, \ldots, \text{Exp}^\gamma_u$, all defined over the same probability space. In each experiment we modify the way the view of $\text{STAT}_u^{\text{IND}}$ is computed, while maintaining the view’s distributions indistinguishable between any two consecutive experiments. For $i = 1, \ldots, \gamma + 1$, experiment $\text{Exp}_u^i$ is defined as follows:

\[
\begin{align*}
\text{Experiment } \text{Exp}_u^i & (1^\gamma, G) \\
(s, h^i, \text{pub}) & \leftarrow \text{Gen}(1^\gamma, G) \\
\text{corr}_u & \leftarrow \text{Corrupt}_u(s) \\
d & \leftarrow \text{STAT}_u^{\text{IND}}(1^\gamma, G, \text{pub, corr}_u, h^i_u) \\
\text{return } d
\end{align*}
\]

Algorithm Gen used in experiment $\text{Exp}_u^i$ is algorithm Gen with the following modification: the last input to $\text{STAT}_u^{\text{IND}}$ is the string

\[
h_u^i = \rho_1 \circ \cdots \circ \rho_{i-1} \circ B_i(k_u^i) \circ B_i(k_{u+1}^i) \circ \cdots \circ B_i(k_{u''}^i),
\]

where $\rho_1, \ldots, \rho_{i-1}$ are random bits chosen independently according to the uniform distribution.

Notice that experiment $\text{Exp}_u^0$ is the same as $\text{Exp}^{\text{IND-0}}_u$. Indeed, the last input to the adversary $\text{STAT}_u^{\text{IND}}$ is the string $h_u^0 = B_i(k_u^0) \circ \cdots \circ B_i(k_{u''}^0)$, corresponding to the key held by class $u$. On the other hand, experiment $\text{Exp}_u^{\gamma+1}$ is the same as $\text{Exp}^{\text{IND-1}}_u$. Indeed, the last input to adversary $\text{STAT}_u^{\text{IND}}$ is the string $h_u^{\gamma+1} = \rho_1 \circ \cdots \circ \rho_{\gamma} \circ r_i$, i.e., a random string of the same length as $k_u$.

We show that, for any $i = 1, \ldots, \gamma$, the adversary’s view in the $i$th experiment is indistinguishable from the adversary’s view in the $(i + 1)$th one. Hence, it follows that also the adversary’s views in experiments $\text{Exp}^{\text{IND-0}}_u$ and $\text{Exp}^{\text{IND-1}}_u$ are indistinguishable.

Assume by contradiction that there exists a polynomial-time algorithm $D_i$ which is able to distinguish between the views in experiments $\text{Exp}^i_u$ and $\text{Exp}_u^{\gamma+1}$ with non-negligible advantage

\[
\text{Adv}_{D_i}^{\gamma+1}(1^\gamma, G) = |\Pr[\text{Exp}^i_u(1^\gamma, G) = 1] - \Pr[\text{Exp}_u^{\gamma+1}(1^\gamma, G) = 1]|.
\]

Assume, without loss of generality, that for infinitely many $\tau$’s,

\[
\Pr[\text{Exp}^i_u(1^\gamma, G) = 1] - \Pr[\text{Exp}_u^{\gamma+1}(1^\gamma, G) = 1] \geq \epsilon(\tau).
\]

We show how to use the algorithm $D_i$ to break the security of the scheme $(\text{Gen}', \text{Der}')$ in the sense of REC-ST. More precisely, we use $D_i$ to construct an algorithm $\text{Pred}$ that, on inputs $(1^\gamma, G, \text{pub}', r, \text{corr}'_u)$, predicts the value $B_i(k_u^0)$ with non-negligible advantage. Then, by Theorem 7.1, using $\text{Pred}$ as an oracle, we construct an adversary $\text{STAT}_u^{\text{REC}}$ which reconstructs the key $k_u^0$ with non-negligible probability. The algorithm $\text{Pred}$, on inputs $(1^\gamma, G, \text{pub}', r, \text{corr}'_u)$, works as follows:

- Sets $\text{pub}' = (\text{pub}', r)$;
- Constructs the private information $\text{corr}_u$ from the private information $\text{corr}'_u$, by removing from $\text{corr}'_u$ the private information held by all classes in $V' \setminus V$;
- Constructs the challenge $h_u$ as follows:
  - Independently and uniformly at random chooses $i$ bits $b_1, \ldots, b_i$;
  - Invokes $\text{Der}'$ on inputs $(1^\gamma, G, u_0, u_j, s_i', \text{pub}')$ to compute the key $k_u^i$, for $j = i + 1, \ldots, \gamma$;
  - Uses the random string $r$ to compute $B_i(k_u^i)$, for $j = i + 1, \ldots, \gamma$;
  - Sets $h_u = b_1 \circ \cdots \circ b_i \circ B_i(k_{u+1}^i) \circ \cdots \circ B_i(k_{u''}^i)$;
- Invokes $D_i$ on input $(1^\gamma, G, \text{pub}, \text{corr}_u, h_u)$. If the output of $D_i$ is 1, then $\text{Pred}$ outputs $h_u$, else, $\text{Pred}$ outputs $1 - b_i$.

Let us evaluate the success probability of $\text{Pred}$. To avoid overburdening the notation, we denote by $D_i(h_u)$ the invocation of $D_i$ on inputs $(1^\gamma, G, \text{pub}, \text{corr}_u, h_u)$. Moreover, we denote by $z$ the string

\[
b_1 \circ \cdots \circ b_{i-1} \circ 1 - B_i(k_u^0) \circ B_i(k_{u+1}^0) \circ \cdots \circ B_i(k_{u''}^0).
\]

Notice that if the bit $b_i$ in the string $h_u$ happens to be equal to $B_i(k_u^i)$, then $h_u = h_u^i$ whereas, if $b_i$ is equal to 1 $- B_i(k_u^i)$, then $h_u = z$. It holds that

\[
\begin{align*}
\text{Prob}[\text{Pred}(1^\gamma, G, \text{pub}', r, \text{corr}_u^i) = B_i(k_u^i)]
&= \text{Prob}[D_i(h_u) = 1 \text{ and } b_i = B_i(k_u^i)] + \text{Prob}[D_i(h_u) = 0 \text{ and } b_i = 1 - B_i(k_u^i)] \\
&= \text{Prob}[D_i(h_u) = 1] \cdot \text{Prob}[b_i = B_i(k_u^i)] + \text{Prob}[D_i(h_u) = 0] \cdot \text{Prob}[b_i = 1 - B_i(k_u^i)] \\
&= \frac{1}{2} \cdot \text{Prob}[D_i(h_u) = 1] \cdot \text{Prob}[b_i = B_i(k_u^i)] + \frac{1}{2} \cdot \text{Prob}[D_i(h_u) = 0] \cdot \text{Prob}[b_i = 1 - B_i(k_u^i)] \\
&\text{(since } b_i \text{ is chosen uniformly at random)} \\
&= \frac{1}{2} \cdot \text{Prob}[D_i(h_u^i) = 1] + \text{Prob}[D_i(z) = 0] \\
&= \frac{1}{2} \cdot \text{Prob}[D_i(h_u^i) = 1] + (1 - \text{Prob}[D_i(z) = 1]).
\end{align*}
\]
It is easy to check that
\[
\text{Prob}[D_i(h_u^{i+1}) = 1] = \frac{1}{2} \text{Prob}[D_i(h_u') = 1] + \frac{1}{2} \text{Prob}[D_i(z) = 1].
\]
Indeed, \(Pr[b_i = B_i(k_u')] = Pr[b_i = 1 - B_i(k_u')] = 1/2\). Thus,
\[
\text{Prob}[D_i(z) = 1] = 2 \cdot [\text{Prob}[D_i(h_u^{i+1}) = 1] - \text{Prob}[D_i(h_u') = 1]].
\]
Therefore, from equality (3) it follows that
\[
\text{Prob}(\text{Pred}(1^r, G', \text{pub}', r, \text{corr}_u') = B_i(k_u'))
\]
\[
= \frac{1}{2} \cdot [\text{Prob}[D_i(h_u') = 1] + (1 - 2 \cdot \text{Prob}[D_i(h_u^{i+1}) = 1] + \text{Prob}[D_i(h_u') = 1])]
\]
\[
= \frac{1}{2} \cdot [1 + 2 \cdot (\text{Prob}[D_i(h_u') = 1] - \text{Prob}[D_i(h_u^{i+1}) = 1])]
\]
\[
= \frac{1}{2} \cdot [1 + 2 \cdot \text{Adv}_i^{\text{IND}}(1^r, G)]
\]
\[
\geq \frac{1}{2} + \epsilon(\tau).
\]
Thus, \(\text{Pred}\) predicts the value \(B_i(k_u')\) with advantage at least \(\epsilon(\tau)\). At this point, we can construct an adversary \(\text{STAT}_i^{\text{REC}}\) which, on inputs \((1^r, G', \text{pub}', r, \text{corr}_u')\), computes \(k_u'\) by using the Goldreich–Levin reconstruction algorithm \(R\). \(\text{STAT}_i^{\text{REC}}\) works as follows:

- Starts running the Goldreich–Levin reconstruction algorithm \(R\);
- Whenever asked for \(B_i(k_u')\), for a certain random string \(r\), it invokes \(\text{Pred}(1^r, G', \text{pub}', r, \text{corr}_u')\) and gives its output to \(R\);
- Outputs whatever \(R\) outputs.

By Theorem 7.1, the Goldreich–Levin reconstruction algorithm \(R\) computes the value \(k_u'\) with probability \(\Omega(\epsilon(\tau))\), since \(\text{Pred}\) predicts \(B_i(k_u')\) with advantage at least \(\epsilon(\tau)\).

Therefore, \(\text{STAT}_i^{\text{REC}}\) computes the key held by class \(u\) with non-negligible advantage. This is a contradiction, since the scheme \((\text{Gen}', \text{Der}')\) is secure in the sense of \(\text{REC–ST}\).

Hence, a distinguisher \(D\) with non-negligible advantage, cannot exist. Therefore, for any \(i = 1, \ldots, \gamma\), the view in the \(i\)th experiment is indistinguishable from the view in the \((i + 1)\)th one. It follows that the adversary’s view in experiment \(\text{Exp}_i^{\text{IND–ST}}\) is indistinguishable from the adversary’s view in experiment \(\text{Exp}_i^{\text{STAT}}\). This concludes the proof.

### 8. An Akl–Taylor key indistinguishable scheme

In Section 4.1 we have shown that the Akl–Taylor scheme is secure in the sense of \(\text{REC–ST}\) under either the random exponent or the fixed exponent RSA assumption. Thus, such a scheme can be used as a building block in the construction of Section 7, in order to obtain a key assignment scheme which is secure in the sense of \(\text{IND–ST}\) under the same assumption.

Notice that the Akl–Taylor scheme cannot be proven secure in the sense of \(\text{IND–ST}\). Indeed, any adversary which attacks class \(u\) knows the key \(k_u\) associated to a class \(v\) child of class \(u\). In order to check if a value \(\rho\) corresponds to the key \(k_u\) the adversary only needs to test whether \(\rho^{2^{|v|}}\) is equal to \(k_u\).

In the following we evaluate the parameters of the scheme obtained by using the construction proposed in Section 7 when the underlying scheme is the Akl–Taylor scheme with the MacKinnon et al. assignment and the random choice of primes. Let \(r\) be a security parameter and let \(\gamma = 2^r\) be the key length. The amount of public information in the resulting scheme corresponds to the \(|V|/(1 + \gamma)\) integers \(t_i\)'s associated to the classes in \(V_{\gamma'}\), in addition to the modulus \(n\) and the \(\gamma\)-bit string \(r\). On the other hand, the construction requires each class to store one secret value, corresponding to its private information. Finally, a class \(u\) which wants to compute the key held by a class \(v \in A_u\) has to perform \(\gamma\) modular exponentiations, to compute the keys \(k_{u_1}, \ldots, k_{u_{\gamma}}\), and \(\gamma\) inner products modulo 2, to compute the GL bits \(B_i(k_u')\)'s. The parameters of the Akl–Taylor key indistinguishable scheme are summarized in the first row of Fig. 4. Assume that, in the MacKinnon et al. assignment, the graph \(G_{\gamma'} = (V_{\gamma'}, E_{\gamma'}) \in \Gamma_{\gamma'}\) is decomposed into \(|V|\) disjoint chains of length \(\gamma + 1\), where each chain contains a class \(u \in V\) along with its associated classes \(u_1, \ldots, u_{\gamma}\). According to the results stated in Section 4.2, the amount of public information required by the Akl–Taylor scheme with the MacKinnon et al. assignment and the modified random prime choice can be reduced to a single prime number, in addition to the modulus \(n\). As explained in Section 4.2, this requires each class to perform on average \(O(|V| \cdot \gamma)\) steps to compute the sequence of primes assigned to the \(V\) chains and \(O(|V| \cdot \gamma)\) products, to compute the prime powers associated to the \(\gamma\) additional classes of each chain. Moreover, the sequence of integers \(t_i\)'s associated to classes in \(V_{\gamma'}\) can be computed by performing \(O(|V|^{2})\) products.

### 9. Conclusions and summary of the results

We have analyzed the Akl–Taylor scheme as well as some of its variants with respect to security and efficiency requirements. Motivated by the fact that the Akl–Taylor scheme is not secure w.r.t. key indistinguishability, we have proposed a general construction to setup a key assignment scheme secure w.r.t. key indistinguishability, given any key assignment scheme secure against key recovery. Such a construction is of independent interest and may be useful for different instantiations.
Fig. 3. Comparison between key assignment schemes which are secure in the sense of REC-ST. Akl–Taylor schemes using the modified random primes choice are not shown.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Akl-Taylor</td>
<td>$</td>
<td>V</td>
<td>+1$</td>
<td>One</td>
</tr>
<tr>
<td>fixed primes</td>
<td></td>
<td></td>
<td>• one division</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• one modular exponent.</td>
<td></td>
</tr>
<tr>
<td>Akl-Taylor</td>
<td>$</td>
<td>V</td>
<td>+1$</td>
<td>One</td>
</tr>
<tr>
<td>random primes</td>
<td></td>
<td></td>
<td>• one division</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• one modular exponent.</td>
<td></td>
</tr>
<tr>
<td>Atallah et al.</td>
<td>$</td>
<td>E</td>
<td>+</td>
<td>V</td>
</tr>
<tr>
<td>[3]</td>
<td></td>
<td></td>
<td>decryption</td>
<td></td>
</tr>
<tr>
<td>TLEBC</td>
<td>$</td>
<td>E^*</td>
<td>$</td>
<td>One</td>
</tr>
<tr>
<td>[4]</td>
<td></td>
<td></td>
<td>pairing eval.</td>
<td>BDH</td>
</tr>
<tr>
<td>TLPBC</td>
<td>$</td>
<td>E^*</td>
<td>$</td>
<td>One</td>
</tr>
<tr>
<td>[4]</td>
<td></td>
<td></td>
<td>One pairing eval.</td>
<td>BDH</td>
</tr>
<tr>
<td>DEBC</td>
<td>$</td>
<td>E</td>
<td>+2</td>
<td>V</td>
</tr>
<tr>
<td>[9]</td>
<td></td>
<td></td>
<td>$\text{dist}(u,v)$ PRF eval.</td>
<td></td>
</tr>
<tr>
<td>TLPBC</td>
<td>$</td>
<td>E^*</td>
<td>$</td>
<td>One</td>
</tr>
<tr>
<td>[4]</td>
<td></td>
<td></td>
<td>pairing eval.</td>
<td>BDH</td>
</tr>
<tr>
<td>BEBC</td>
<td>$4</td>
<td>V</td>
<td>+1$</td>
<td>One</td>
</tr>
</tbody>
</table>

Fig. 4. Comparison between key assignment schemes which are secure in the sense of IND–ST. Akl–Taylor schemes using the modified random primes choice are not shown.

Figs. 3 and 4 show comparisons between our constructions and previous proposals for key assignment schemes secure in the sense of REC–ST [3,4] and in the sense of IND–ST [3,4,8], respectively. The comparisons take into account the amount of public and private information, the number and the type of operations required to perform key derivation, and the computational assumption. We denote by $\text{dist}(u,v)$ the length of the shortest path from $u$ to $v$.

Atallah et al. [3] proposed a first construction based on pseudorandom functions (PRF) and a second one requiring PRF and a CCA–secure symmetric encryption scheme. Ateniese et al. proposed two constructions, which we refer to as the Two-Level Encryption Based Construction (TLEBC) and the Two-Level Pairing Based Construction (TLPBC). The TLEBC makes use of a symmetric encryption scheme secure either against plaintext recovery (PR–P1–CO), or against plaintext indistinguishability (IND–P1–CO). The TLPBC [4] makes use of pairings and assumes the intractability of either the Bilinear Diffie–Hellman problem (BDH) or the Bilinear Decisional Diffie–Hellman problem (BDDH). A different construction, called the Dynamic Encryption Based Construction (DEBC), making use of an IND–P1–CO symmetric encryption scheme, was proposed in [8]. Finally, the Broadcast Encryption Based Construction (BEBC) proposed in [8] makes use of a public-key broadcast encryption scheme and assumes the intractability of the $|V|\cdot\text{Bilinear Decisional Diffie–Hellman Exponent problem (}|V|\cdot\text{BDDHE})$.

References