Харківський національний університет імені В.Н. Каразіна
Факультет математики і інформатики
Кафедра прикладної математики

# Кваліфікаційна робота

рівень ***магістр***

## на тему ***«Схема Акла-Тейлора і аналіз її безпеки»***

*Виконав:* студент групи МП62 II курсу (другий магістерський рівень), спеціальності 113 “Прикладна математика” освітньо-професійної програми “Прикладна математика”
**Картишев Є.С.**

*Керівник:* доктор фіз.-мат. наук, доц., професор кафедри прикладної математики
**Ігнатович С.Ю.**

*Керівник:* Ricercatore, TD-a
**Roberto Civino (L’Aquila)**

*Рецензент:* доктор техн. наук, доц., професор кафедри комп.-інтегр. технологій авт. та робототехн., ХНУРЕ
**Ромашов Ю.В.**

Харків — 2024 рік

V. N. Karazin Kharkiv National University
Faculty of Mathematics and Computer Science
Department of Applied Mathematics

# Master thesis

on the topic «***Akl-Taylor scheme and its security analysis***»

*Author:* student of AM62 group of II year
(Master level)
speciality 113
"Applied Mathematics"
educational and professional program
"Applied Mathematics"
**Kartyshev Y.**

*Supervisor:* D.Sc. in physics and mathematics,
Professor of the Department of Applied Mathematics
**Ignatovich S.Yu.**

*Supervisor:* Ricercatore,
TD-a
**Roberto Civino (L'Aquila)**

*Reviewer:* D.Sc. in technical science,
Professor,
Kharkiv National University of Radio Electronics
**Romashov Yu.V.**

Kharkiv — 2024

# Анотації

**Картишев Єгор Сергійович. Схема Акла-Тейлора і аналіз її безпеки.**

У цій роботі ми розглянемо схему Акла-Тейлора призначення ключів, зрозуміємо, як вона влаштована і які існують різновиди її реалізації. Схеми призначення ключів полегшують розподіл ключів користувачам, дозволяючи їм розшифровувати та отримувати доступ до певної інформації. Як правило, користувачі організовані в ієрархічну структуру відповідно до своїх повноважень, що дозволяє користувачеві з вищими повноваженнями отримувати ключі від усіх підлеглих користувачів з нижчими повноваженнями. Ми обговоримо основні атаки на системи, що використовує цю схему і захищеність нашої системи стосовно цих атак.

**Ключові слова:** ієрархічне призначення ключа, алгоритм RSA, незначна функція.

# Annotation

**Kartyshev Yehor. Akl-Taylor scheme and its security analysis.**

In this paper we will consider the Akla-Taylor scheme for key assignment, understand how it is structured and what types of its implementation exist. Key assignment schemes facilitate the distribution of keys to users, allowing them to decrypt and access specific information. Typically, users are organized into a hierarchical structure according to their authority, enabling a user with higher authority to derive the keys of all subordinate users with lower authority. We will discuss the main attacks on systems using this scheme and the security of our system against these attacks.

**Keywords:** hierarchical key assignment, RSA algorithm, negligible function.

# Contents

1.Introduction 5

2.Our goals 7

3.Describing of the model 9

4.The Akl-Taylor scheme 15

5.The security of Akl-Taylor scheme 17

6.Conclusions 22

7.Bibliography 23

# 1.Introduction

In various applications where users and resources can be organized into partially ordered hierarchies, hierarchical key assignment schemes are extensively employed to establish secure access control policies. The natural organization provided by hierarchies, reflecting users' roles within an organization, makes them a fundamental component across diverse domains (e.g., database management systems, computer networks, operating systems, military, government communications). Hierarchies play a crucial role in scenarios modeled using Role Based Access Control (RBAC).

In 1983, Akl and Taylor introduced a novel perspective by proposing the integration of cryptographic techniques to fortify access control mechanisms within hierarchical structures [5] . Their innovative approach materialized in a hierarchical key assignment scheme, where each class received a dedicated encryption key. This key, coupled with specific public parameters, empowered the calculation of keys assigned to all subordinate classes in the hierarchy.

The Akl–Taylor scheme, celebrated for its simplicity and versatility, has seen widespread adoption in enforcing access control across diverse domains. Its application extends to varied contexts, including mobile agent environments and XML documents. Beyond its direct implementation, the Akl–Taylor scheme has proven instrumental as a foundational template for conceiving key assignment schemes tailored to enforce more expansive access control policies. These encompass scenarios featuring transitive and anti-symmetrical exceptions, as well as those grappling with time-dependent constraints.

Furthermore, the impact of the Akl–Taylor scheme transcends specific ap-

plications, as evidenced by its incorporation into the design of broadcast encryption protocols. Noteworthy contributions by Asano and Attrapadung and Kobara underscore the scheme's adaptability and utility within the dynamic landscape of cryptographic endeavors [6] .

# 2.Our goals

Within the scope of this thesis, we conduct a thorough examination of the Akl–Taylor scheme and its variations, focusing on both security and efficiency aspects.

In terms of security, we delve into the Akl–Taylor scheme, aligning our analysis with the definitions in. Special attention is given to the careful selection of public parameters to ensure the scheme's instances remain secure against key recovery under the RSA assumption. Notably, we explore the impact of hierarchy size on key derivation complexity in the Akl–Taylor scheme. To address this, MacKinnon et al. propose an alternative generation of public values, while Harn and Lin introduce a variant designed for more efficient key derivation in broader and shallower hierarchies. Our investigation confirms the security of both the MacKinnon et al. and Harn–Lin variants against key recovery. Extending our analysis, we examine the security of the reduced Akl–Taylor assignment, providing formal proofs for its application in schemes covering diverse access control policies.

Moreover, we tackle the issue of designing an Akl–Taylor-based scheme that ensures security concerning key indistinguishability. Proposing a general construction, we introduce an independent solution that yields a key assignment scheme offering security with respect to key indistinguishability, leveraging any key assignment scheme guaranteeing security against key recovery.

Efficiency Considerations: Turning our attention to efficiency considerations, we demonstrate the continued security of the Akl–Taylor scheme, even when disclosing only a fraction of the public information — as minimal as

a single prime number. However, this heightened security comes at the cost of a more resource-intensive key derivation process. Thus, we bring to light a tradeoff between the size of public information and the complexity of key derivation.

Furthermore, we emphasize the superior performance of Akl–Taylor-based schemes compared to other existing provably-secure schemes. Particularly noteworthy is their efficiency in areas beyond key derivation complexity. Akl–Taylor-based schemes with time constraints demand minimal public information, showcasing heightened efficiency compared to other time-dependent, provably-secure schemes.

# 3.Describing of the model

Contemplating a group of users organized into distinct classes, termed security classes, is fundamental in delineating the structure of an organization's access control system. Each security class serves as a representation for an individual, a department, or a specific user group within the organizational framework.

To establish the hierarchical structure among these classes, a binary relation denoted as $\preceq$ is introduced. This relation serves to partially order the set of classes V based on the inherent authority, position, or power associated with each class within the organizational context. The relationship $\preceq$ provides a systematic means of representing the relative influence or ranking of different classes within the defined hierarchy.

The structure $(V, \preceq)$, referred to as a partially ordered hierarchy or poset, delineates the hierarchical relationships among the classes. In this hierarchy, the notation $v \preceq u$ signifies that users in class $u$ have access to the data of class $v$. It's evident that u can access its own data, leading to the assertion that $u \preceq u$ for any $u$ in $V$.

To formalize the access relationships we consider the following definitions proposed in [4] :

**Definition 1.** *Let $A_u$ be the set of nodes to which node $u$ has access, denoted as $A_u = \{v \in V : v \preceq u\}$, for any $u$ in $V$.*

This set captures the nodes accessible from a given node $u$ within the partially ordered hierarchy. Let's

**Definition 2.** *Let $F_u = \{v \in V : v \notin A_u\}$*

The hierarchy $(V, \preceq)$ finds representation in the directed graph $G^* = (V, E^*)$, where each class corresponds to a vertex, and there exists an edge from class $u$ to class $v$ if and only if $v \preceq u$. Introducing another graph, $G = (V, E)$, as the minimal representation of $G^*$, involves constructing a directed acyclic graph. This minimal representation is derived from the transitive and reflexive reduction of $G^*$, preserving the transitive and reflexive closure. In simpler terms, G shares the same transitive and reflexive closure as $G^*$, meaning there is a path from $u$ to $v$ in $G$ if and only if there is an edge $(u, v)$ in $E^*$. This minimal representation ensures a concise depiction of the relationships while eliminating redundancy.

**Definition 3.** *Let's define an algorithm Gen. It will generate information and it is probabilistic polynomial time. Gen has as input: private information* $1^\tau$ *and graph* $G = (V, E)$*. Output will consists of:* $s_u$ *- private information,* $k_u$ *- private key* $\forall u \in V$*, pub - public information.*

In other words we have: $Gen(1^\tau, G) = (s, k, pub)$ , where $s, k$ are corresponding sequences.

**Definition 4.** *Let's define an algorithm Der and it is deterministic polynomial-time. Der has as input: private information* $1^\tau$ *and graph* $G = (V, E)$*, class* $u \in V$*,* $v \in A_u$*,* $s_u$ *- private information that linked with class* $u$*, pub - public information that generates by Gen. This algorithm will output the key* $k_v$ *that linked with class* $v$

In other words we have: $Der(1^\tau, G, u, v, s_u, pub) = k_v$

**Definition 5.** *The system* $(Gen, Der)$ *we will call a hierarchical key assignment*

Let's define a static adversary which is attacking the class $v \in V$ and let's assume that our static adversary is able to collect the information from any

class $u$ that has no access to class $v$. Let's define an algorithm that will collect this information:

**Definition 6.** *$Corrupt_v$ is an algorithm with an input that consists of private information $s$ that is generated by $Gen$, the collection of $s_u$, where $u \in F_v$. Let's denote the input of this algorithm by $corr_v$*

Later in this text we will use the definition of negligible function:

**Definition 7.** *The function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if $\forall$ constant $c > 0$ $\exists$ an integer $n :$ $f(x) < x^{-c}$ for all $x > n$*

Let's define a static adversary attack $Rec - St$:

**Definition 8.** *(Rec-St) Assume $G = (V, E)$ - graph, $(Gen, Der)$ - hierarchical key assignment scheme, let $Stat_v^{Rec}$ - static adversary that attacks class $v$. Let's define such an experiment:*
*Experiment:*
$Exp_{Stat_v}^{Rec}(1^\tau, G)$
$(s, k, pub) \leftarrow Gen(1^\tau, G)$
$corr_v \leftarrow Corrupt_v(s)$
$k_v^* \leftarrow Stat_v^{Rec}(1^\tau, G, pub, corr_v)$
$return\ k_v^*$
*Let's define an advantage of $Stat_v^{Rec}$ as $Adv_{Stat_v}^{Rec}(1^\tau, G) = Pr[k_v^* = k_v]$.*

**Definition 9.** *We said the the scheme is secure in the sense of $Rec - St$ if $\forall$ graph $G = (V, E)$ , $v \in V$, the function $Adv_{Stat_v}^{Rec}(1^\tau, G)$ is negligible for $\forall$ static adversary with polynomial time complexity in $\tau$ - $Stat_v^{Rec}$*

Let's define a static adversary attack $Ind - St$:

**Definition 10.** *Assume $G = (V, E)$ - graph, $(Gen, Der)$ - hierarchical key assignment scheme, let $Stat_v^{Ind}$ - static adversary that attacks class $v$. Let's*

*define two experiments:*

*Experiment:*
$Exp^{Ind-1}_{Stat_v}(1^\tau, G)$
$(s, k, pub) \leftarrow Gen(1^\tau, G)$
$corr_v \leftarrow Corrupt_v(s)$
$d \leftarrow Stat^{Ind}_v(1^\tau, G, pub, corr_v, k_v)$
$return\ d$

*Experiment:*
$Exp^{Ind-0}_{Stat_v}(1^\tau, G)$
$(s, k, pub) \leftarrow Gen(1^\tau, G)$
$corr_v \leftarrow Corrupt_v(s)$
$\rho \leftarrow (0,1)^{length(k_v)}$
$d \leftarrow Stat^{Ind}_v(1^\tau, G, pub, corr_v, \rho)$
$return\ d$

Similarly with $Rec - St$ we should define an advantage of $Stat^{Ind}_v$ and define the security of the system in sense of $Ind - St$:

**Definition 11.** *Advantage of* $Stat^{Ind}_v$ *:*$Adv^{Ind}_{Stat_v}(1^\tau, G) =$
$|Pr\{Exp^{Ind-1}_{Stat_v}(1^\tau, G) = 1\} - Pr\{Exp^{Ind-0}_{Stat_v}(1^\tau, G)) = 1\}|$
*We said the the scheme is secure in the sense of* $Ind - St$ *if* $\forall$ *graph* $G = (V, E)$ , $v \in V$, *the function* $Adv^{Ind}_{Stat_v}(1^\tau, G)$ *is negligible for* $\forall$ *static adversary with polynomial time complexity in* $\tau$ *-* $Stat^{Ind}_v$

Later in the text we will use generators that based on RSA. Let's define these generators:

**Definition 12.** $K^{ran}_{RSA}(1^\tau)$
*Choose randomly two distinct prime numbers* $p$, $q$ *with the size of* $\tau$ *bits*

$n \leftarrow p * q$

$\phi(n) \leftarrow (p-1)(q-1)$

*choose* $e$ *from* $\mathbb{Z}^*_{\phi(n)}$

$d \leftarrow e^{-1}$ *mod* $\phi(n)$

*return* $((n,e),(n,p,q,d))$

**Definition 13.** $K^{fix}_{RSA}(1^\tau, e)$

*We repeat this:*

*Choose randomly two distinct prime numbers* $p$, $q$ *with the size of* $\tau$ *bits*

*until* $gcd(p-1,e) = gcd(q-1,e) = 1$

$n \leftarrow p * q$

$\phi(n) \leftarrow (p-1)(q-1)$

*choose* $e$ *from* $\mathbb{Z}^*_{\phi(n)}$

$d \leftarrow e^{-1}$ *mod* $\phi(n)$

*return* $((n,e),(n,p,q,d))$

Now we should formulate an experiment, let's call the generator $H$, where $H$ is $K^{fix}_{RSA}(1^\tau, e)$ or $K^{ran}_{RSA}(1^\tau)$

**Definition 14.** $Exp^H_B$

$((n,e),(n,p,q,d)) \leftarrow H$

$x \leftarrow \mathbb{Z}^*_\ltimes$

$y \leftarrow x^e$ *mod* $n$

$x' \leftarrow B(n,e,y)$

*if* $x' = x$*:*

*return 1*

*else*

*return 0*

*With an advantage* $Adv^H_B = Pr\{Exp^H_B = 1\}$

Now let's formulate two statements [2, 4]:

**Theorem 1.** *(Random Exponent RSA Assumption)*
*Advantage* $Adv_B^{K_{Rsa}^{ran}}((1^\tau))$ *is negligible, for* $\forall$ *probabilistic method B with time polynomial complexity in* $\tau$

So the probability $Pr\{Exp_B^{K_{Rsa}^{ran}} = 1\}$ is a negligible regardless of which algorithm B we will choose. We'll need this fact later when we'll talk about the security of the scheme.

**Theorem 2.** *(Fixed Exponent RSA Assumption)*
*Assume we have a set of odd numbers* $O$, $\forall e \in O$ *advantage* $Adv_B^{K_{Rsa}^{fix}}((1^\tau))$ *is negligible then for* $\forall$ *probabilistic method B with time polynomial complexity in* $\tau$

This fact is similar to the previous fact with the only difference that it is formulated for the fixed encryption exponent e.

# 4.The Akl-Taylor scheme

Now we are able to define the Akl-Taylor scheme [5]. At first we should define the hierarchical key assignment for this scheme. Let's define algorithm $Gen(1^\tau, G)$:

(1) Let's choose two different prime numbers: $p$,$q$, such that $p \neq q$ , $p$ and $q$ have $\tau$ bit size.

(2) $\forall v \in V$ define $t_v$ such that $t_u$ divides $t_v \Leftrightarrow v \in A_u$

(3) Define *pub* as sequence of the information that we defined in the previous step.

(4) Let's choose random $k_0$ - secret value, such that $1 < k_0 < n$

(5) Let's choose the private information $s_v$ and the encryption key $k_v$: $s_v = k_v = k_0^{t_v} \bmod n$

(6) Define $s$ and $k$ as private information that we define in the previous step

(7) Output $(s, k, pub)$

Let's define algorithm $Der(1^\tau, G, u, v, s_u, pub)$: From *pub* we have $t_v$,$t_u$

Let's compute $s_u^{t_v/t_u} \bmod n = (k_0^{t_0})^{t_v/t_u} \bmod n = k_v$

We defined the hierarchical key assignment, but we should decide how to apply the (2) item from the list. There are two options:

**Definition 15.** *(The Akl–Taylor assignment) $\forall v \in V$ are choosing different $p_v$ and define $t_v$:*

$t_v = 1$ *if* $A_v = V$ *and* $t_v = \prod_{u \notin A_v} p_u$ *otherwise.*

**Definition 16.** *(The MacKinnon assignment) Let's divide graph $G$ into chains with total order and compare to each chain a prime number so all the primes are different. For all $v \in V$ compare $n_v = p^i$ where $i$ is the order of $v$*

*in the relevant chain that assign to p. For* $\forall\ v \in V$ *define* $t_v$*:*
$t_v = 1$ *if* $V = A_v$ *and* $t_v = lcm_{u \notin A_v} n_u$ *otherwise*

After we describe the second part of *Gen* we should describe how we will choose our primes. There are two options:

**Fixed prime choice**:

Let's define the set of $l$ distinct prime numbers that $> 2$ : $P_l = (p_1, p_2, ..., p_l)$
(1) In the Akl-Taylor assignment let's sort $V$: $\{u_1, u_2, ..., u_{|V|}\}$. Let's compare for $\forall u_i \in V$ the prime number $p_i \in P_{|V|}$
(2) In the MacKinnon et al. assignment let's consider the minimal dividing of $G$ to the total ordered chains. Assume that the amount of chains is $m$. Let's compare for $\forall p_i \in P_m$ class $u_i$, where $u_i$ is the first item in the $i$-th chain

**R-Random prime choice**:

Assume $n$ = RSA module, $R = \{R_n\}_n$ - family of sets of integer. (1) In the Akl-Taylor scheme we compare for $\forall$ class from $V$ a prime number from $R_n$.
(2)In the MacKinnon et al. assignment we compare for $\forall$ chain in the minimal dividing of the graph $G$ a prime number from $R_n$

# 5.The security of Akl-Taylor scheme

Now we are able to deal with security of the Akl-Taylor scheme. First of all let's talk about the Akl-Taylor assignment of this scheme [2, 4].

**Theorem 3.** *Assume $G = (V, E)$ - the partially ordered hierarchy, The scheme of the Akl-Taylor assignment with the fixed prime choice is secure to $Rec - St$.*

Proof

Assume that there is a static adversary $Stat_{u_i}$ that can find out $k_{u_i}$ that assign to the class $u_i$ with non-negligible advantage. Let's construct a polynomial time algorithm $B(n, e, y)$, where $e$ is choose from $P_{|V|}$ and $e = p_i$, $n$ is generated by $K^{fix}_{RSA}(1^\tau, G)$. Let's describe how we will construct the algorithm $B(n, e, y)$:

Now we should construct the input for $B(n, e, y)$: compare $p_j$ to class $u_j$, given what we knows that $p_i = e$. Then, as we write before, $t_{u_j} = \prod_{l \notin A_{u_j}} p_l$, and if $A_{u_j} = V$ then $t_{u_j} = 1$. The public information that we got from the last step we combine to the sequence *pub*, augmented by the inclusion of the value n. Let's compute key $k_v = y^{t_v/p_i} \bmod n$ for $\forall v \in F_{u_i}$. We can do this, because $u_i \notin A_{u_j}$ so $p_i | t_v$. The sequence of $k_v$ that we computed yield a sequence of $corr_{u_i}$.

Then after we define the input, let's denote $k_{u_i} = Stat_{u_i}(1^\tau, G, pub, corr_{u_i})$. Let's pick such integer numbers $\alpha, \beta$ that $\alpha * p_i + \beta * t_{u_i} = 1$. There are such numbers, because $gcd(p_i, t_{u_i}) = 1$, so we can find $\alpha, \beta$ by using Extended Euclidean Algorithm. Now we can compute $x$ just by computing $y^\alpha * k^\beta_{u_i} \bmod$

$n$. We will get $x$, because $y^{\alpha} * k_{u_i}^{\beta} \bmod n = x^{\alpha * p_i} * x^{\beta * t_{u_i}} \bmod n = x^{\alpha * p_i + \beta * t_{u_i}} \bmod n = x$.

Now it's just left to consume that $Adv_B^{K_R SA}(1^{\tau}, e) = Adv_{Stat_{u_i}(1^{\tau})}$ and as we said before $Adv_B^{K_R SA}$ is negligible, so $Adv_{Stat_{u_i}(1^{\tau})}$ is negligible, but we assumed that $Adv_{Stat_{u_i}}(1^{\tau})$ is non-negligible. We got a contradiction.

So we have proved that the Akl-Taylor assignment with the fixed prime choice is secure to $Rec - St$. Now let's prove that the scheme with the same hierarchical key assignment with the random prime choice is secure in the sense of $Rec - St$.

**Theorem 4.** *The scheme with the Akl-Taylor assignment with the random prime choice is secure to* $Rec - St$.

Proof

For this theorem we will need two lemmas. Let's formulate the first of them:

**Lemma 1.** *Let* $p, q$ *are two distinct prime numbers with the bit size* $\tau$, $n = p * q$. *So the Euler function satisfies the following inequality:*

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

Proof

Our prime numbers $p, q$ have bit length $\tau$, so $p, q > 2^{\tau - 1}$ thus,

$\phi(n) = (p-1) * (q-1) > (2^{\tau - 1} - 1) * 2^{\tau - 1}$, because $p, q$ are distinct. Thus, $\phi(n) > 2^{2\tau - 2} - 2^{\tau - 1} = 2^{2\tau - 2} - 2^{\tau}$

For the next lemma we will need some knowledge from number theory. First of all let's recall the Prime Number Theorem:

**Theorem 5.** *Let* $\pi(x)$ *- number of prime numbers that* $\leq x$. *Then* $\pi(x) \sim x / \ln(x)$ *and* $\forall x > 17 : \pi(x) > x / \ln(x)$.

Next fact:

**Theorem 6.** *Let $\omega(x)$ be the number of different prime factors of $x$. Then the function $\omega(x)$ has normal order $\log(\log x)$. Besides, the function $\omega(\phi(n))$ has normal order $(\log(\log n))^2/2$ and the following inequality holds:*
$(1-\epsilon) * (\log(\log n))^2/2 < \omega(\phi(n)) < (1+\epsilon) * (\log(\log n))^2/2$.

Now we are able to formulate the lemma:

**Lemma 2.** *Let $A_w = \{a < w : a \text{ is prime}\}$, $B_w = \{a < w : a \text{ is prime and } gcd(a, \phi(n)) = 1\}$. Let $w = 2^{2\tau-2} - 2^{\tau}$ and $n$'s bitlength $\leq 2\tau$. Then $|A_w - B_w|/|A_w|$ - negligible function in $\tau$.*

Proof

First of all let's note that $|A_w| = \pi(w)$ and $|A_w| - |B_w| \leq \omega(\phi(n))$:

$$\frac{|A_w| - |B_w|}{|A_w|} \leq \frac{\omega(\phi(n))}{\pi(w)} < (1+\epsilon) * \frac{(\log(\log n))^2}{2} * \frac{\ln w}{w}$$

Notice that $\log(\log n) < \log(2\tau)$ , because $n < 2^{2\tau}$, then $\log(n) < 2\tau$, then $\log\log(n) < \log(2\tau)$. Notice that $w > 2^{2\tau-3}$ and $\dfrac{\ln w}{2} < \tau$. Then,
$|A_w| - |B_w|/|A_w| < (1+\epsilon) * \dfrac{(\log(2\tau))^2 * \tau}{2^{2\tau-3}}$ - negligible.

Now we are able to prove the theorem: As in the previous proof of the security we will assume that there is a static adversary $B$ which is able to compute with non-negligible advantage the key $k_u$. We will construct an algorithm $B(n, e, y)$, where $n, e$ are chosen from the generator $K^{ran}_{RSA}(1^{\tau})$:

We got $e$ from $K^{ran}_{RSA}(1^{\tau})$ and if $e$ is not prime and not satisfy an inequality $3 < e < \omega$ - stop and try to choose $e$ again. So if $e$ is prime let's set our $p_u = e$ and choose a distinct $p_v$ for every $v \in V, v \neq u$ that is prime that $\neq e, \in [3, w]$. Then we compute $t_v = \prod_{a \notin A_v} p_a$, but if $t_v = 1$ if $A_v = V$. Unite these

numbers into the sequence $pub$, along with $n$. Now $\forall v \in F_u$ let's compute the key $k_v = y^{t_v/p_u} \bmod n$, unite these keys into the sequence $corr_u$. Denote $k_u = Stat_u(1^\tau, G, pub, corr_u)$.

Again we need to use the Extended Euclidean Algorithm to compute $\alpha, \beta$ such that $p_u * \alpha + t_u * \beta = 1$, because as in the previous proof: $gcd(p_u, t_u) = 1$. It's just left to compute: $y^\alpha * k_u^\beta \bmod n = x^{p_u * \alpha} * x^{t_u * \beta} \bmod n = x$. Now we got that $Adv_{Stat_u}(1^\tau)$ is non-negligible. Let's denote $Adv_{Stat'_u}(1^\tau)$ be the advantage with restriction that $gcd(p_u, \phi(n)) = 1$. If $Adv_{Stat'_u}(1^\tau)$ is negligible, then $Stat_u$ is breaking our scheme if only $gcd(p_u, \phi(n)) \neq 1$, but accordingly the Lemma 2, the probability that $gcd(p_u, \phi(n)) \neq 1$ is negligible, then $Adv_{Stat_u}$ - negligible, then we got a contradiction. So, we got that $Adv_{Stat'_u}(1^\tau)$ is non-negligible.

Let's note that:

$Adv_B^{K^{ran}_{RSA}}(1^\tau) = \text{Probability}(e \text{ is prime and } gcd(e, \phi(n)) = 1) * Adv_{Stat'_u}(1^\tau) =$

$$= \frac{\pi(w) - \omega(\phi(n))}{\phi(\phi(n))} * Adv_{Stat'_u}(1^\tau) \geq \frac{\frac{w}{\ln w} - \omega(\phi(n))}{\phi(n)} * Adv_{Stat'_u}(1^\tau) \geq$$

$$\geq \frac{w - \ln w * \omega(\phi(n))}{\ln w * \phi(\phi(n))} * Adv_{Stat'_u}(1^\tau) \geq \frac{2^{2\tau-2} - 2^\tau - 2\tau * \omega(\phi(n))}{2\tau * 2^{2\tau}} * Adv_{Stat'_u}(1^\tau) \geq$$

$$\geq \frac{2^{2\tau-2} - 2^\tau - \tau * (1+\epsilon) * (\log\log n)^2}{2\tau * 2^{2\tau}} * Adv_{Stat'_u}(1^\tau) \geq$$

$$\geq \frac{2^{2\tau-2} - 2^\tau - O(\tau^3)}{2\tau * 2^{2\tau}} * Adv_{Stat'_u}(1^\tau) \geq \frac{1}{c * \tau} * Adv_{Stat'_u}(1^\tau)$$

This function is non-negligible thus, $Adv_B^{K^{ran}_{RSA}}(1^\tau)$ is non-negligible. We got a contradiction.

**Theorem 7.** *Let $G = (V, E)$ - partially ordered hierarchy, let $m$ be a*

*number of total ordered chains in the minimal dividing of G.The scheme with the MacKinnon et al.assignment with prime choice in $P_m$ is secure to $Rec = St$.*

Proof

Let $e = p_i$. As usual let's assume that there is a static adversary $Stat_u$ that is able to compute with non-negligible advantage the key $k_u$ that is compared with the class $u$, where $u$ is the $l$-th node in the $i$-th chain. As before we will construct an algorithm $B(n, e, y)$ where $n$ is choosing accordingly to the generator $K_{RSA}^{fix}$ and with $B(n, e, y)$ we will compute $x \in \mathbb{Z}_n^*$, such that $y = x^e \bmod n$. Let's construct this algorithm $B$:

At first we will define an output: $\forall$ integer $j \in [1, m]$ we will compare $p_j$ to class $u_j$ with respect to $p_i = e$. Then $\forall v \in V$ we will compute $n_v = p_j^l$, where $v$ - $l$th node in the $j$-th chain. Now we can compute $t_v = lcm_{a \notin A_v} n_a$, but if $A_v = V$, then $t_v$ will equal to 1. Now let's extract this values $t_v$ to a sequence *pub*, along with value $n$.

After this we need to define our $corr_u$, for this we should define $k_v$ for every $v \in F_u$. Let's do this: $t_v = (y^{1/e^l})^{t_v} \bmod n$, so the $corr_u$ with be the sequence of this keys.

Let $Stat_u(1^\tau, G, pub, corr_u) = k_u$. As before we will use the Extended Euclidean Algorithm to find such $\alpha, \beta$ that $\alpha * p_u + \beta * \frac{t_u}{e^{l-1}} = 1$. Let's explain why $\frac{t_u}{e^{l-1}}$ is integer: let's denote $u'$ the $(l-1)$-th node of the $i$-th chain, then $u$ has no access to $u'$ thus, $p_i^{l-1} | t_u$, then $e^{l-1} | t_u$.

Ok, so we can find such $\alpha, \beta$, then let's compute $y^\alpha * k_u^\beta \bmod n = x^{p_u * \alpha} * x^{\frac{t_u}{e^{l-1}} * \beta} \bmod n = x$.

It's just left to consume that $Adv_B^{K_{RSA}^{fix}}(1^\tau, G) = Adv_{Stat_u}$, and we got that the last advantage is non-negligible, so $Adv_B^{K_{RSA}^{fix}}(1^\tau, G)$ is non-negligible too. We got a contradiction.

# 6.Conclusions

We have figured out what a hierarchical key assignment is and defined different types of static adversary. Thanks to this, we were able to define the Akl-Taylor scheme and considered several ways to build it. After defining the scheme, it was necessary to check its security, which we successfully did. We defined different types of attacks and checked the stability of the scheme in its variations to various attacks. Thus, we demonstrated the security of the scheme in our implementation. In the future, we can try to come up with another implementation and examine its security. Also note that everything we did took place in the context of the ring of integers, it will be interesting to consider how basic definitions can be implemented on two rings and how the results will differ.

# Bibliography

[1] Selim G. Akl and Peter D. Taylor, Queen's University, "Cryptographic Solution to a Problem of Access Control in a Hierarchy", ACM Transactions on Computer Systems (TOCS), Volume 1, Issue 3, Pages 239 - 248, 1983

[2] Paolo D'Arco, Alfredo De Santis, Anna Lisa Ferrara, Barbara Masucci, "Variations on a theme by Akl and Taylor: Security and tradeoffs", Theoretical Computer Science, Volume 411, Issue 1, Pages 213-227, 2010

[3] Stephen J.Mackinnon, Peter D. Taylor, Henk Meijer, and Selim G. Akl, "An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy", IEEE Transactions on Computers, vol. 34, No. 9, 1985

[4] Paolo D'Arco, Alfredo De Santis, Anna Lisa Ferrara1, and Barbara Masucci, "Security and Tradeoffs of the Akl-Taylor Scheme and Its Variants", Mathematical Foundations of Computer Science, pp 247–257, 2009

[5] S. Akl and P. Taylor, "Cryptographic solution to a problem of access control in a hierarchy", ACM Transaction on Computer Systems, Vol. 1, n. 3, pp. 239-248, 1983.

[6] N. Attrapadung, K. Kobara, H. Imai, "Sequential Key Derivation Patterns for Broadcast Encryption and Key Predistribution Schemes", ASIACRYPT 2003