Anonymity for Key-Trees with Adaptive Adversaries[⋆]

Michael Beye[1] and Thijs Veugen[1,2]

[1] Information Security and Privacy Lab, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands
m.r.t.beye@tudelft.nl
[2] Security Group, TNO, The Netherlands
thijs.veugen@tno.nl

Abstract. Hash-lock authentication protocols for Radio Frequency IDentification (RFID) tags incur heavy search on the server. Key-trees have been proposed as a way to reduce search times, but because partial keys in such trees are shared, key compromise affects several tags. Buttyán [4] and Beye and Veugen [3] devised trees to withstand such attacks, but assumed adversaries to be non-adaptive, without access to side-channel information. We illustrate how in practice, side-channel information can be used to attack the system. We also describe adaptive attacks that are easy to mount and will significantly reduce tag anonymity. Theoretical analysis of the implications on anonymity in key-trees leads to new requirements and a new tree construction. Simulation is used to test its performance, the results showing an improved resistance to adaptive attacks.

Keywords: RFID, Hash-lock protocol, key-tree, anonymity, anonymity set, adaptive adversaries.

1 Introduction

We consider the problem of authenticating many Radio Frequency IDentification (RFID) tags through hash-lock protocols, in an efficient way. The tags are authenticated towards the reader through a challenge-response mechanism. Each tag authenticates itself using some secret key combined with a random value. To authenticate the tag, the reader will have to check the keys of all tags combined with all possible random values, in order to find a match. Since this task is very intensive for the reader, a key-tree is used. Each leaf of the tree represents a tag, and each edge corresponds to a specific key. Every tag is assigned the keys that lie on its path from the root of the tree (see Fig. 1). During the authentication protocol, a tag is authenticated step by step, i.e. edge by edge, such that the computational load of the reader, and thus the total authentication time, is lowered.

[⋆] Part of this research was performed at TNO for a master's thesis for the University of Utrecht (UU). Special thanks go to Gerard Tel (UU) for his advice, and to Harry Fluks (TNO) for his work on the simulation code.

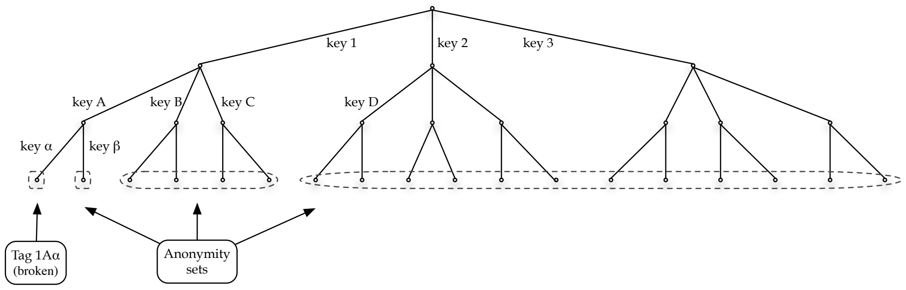

Fig. 1. Key-tree with a single broken tag

However, the authentication mechanism should still remain secure. If hardware level tampering is taken into account, keys that were assigned to compromised tags can become known to the adversary. Because partial keys are shared between neighboring tags in the tree, several additional tags may be partially broken as well. How to construct the tree such that the impact to an average tag's anonymity will be minimal in case of one or more compromises?

In existing work on tree optimization [4, 3], adversaries are assumed not to mount adaptive or side-channel attacks. However, we argue that in practice side-channel information may be readily available, and we show how adaptive attacks on the system can be mounted with minimal effort.

The main contribution of this paper is twofold. First, the effects of adaptive and side-channel attacks on anonymity in key-trees are studied and distilled into a new tree optimization problem. Because this problem is diametrically opposed to Buttyán's original optimization problem, a hybrid defense strategy is devised and tested, to provide protection from both naive *and* adaptive / side-channel attacks.

The layout of this paper is as follows: Section 2 will outline related work, with a brief explanation of most relevant concepts. Section 3 will focus on side-channel information, adaptive attackers and targeted attacks. Section 3.1 considers the impact of such attacks on key-tree anonymity, and proposes a novel type of tree as defence. Section 4 will evaluate the performance of this new construction by means of simulations, and finally, conclusions are drawn in Section 5.

2 Related Work

Molnar was the first to propose using a *tree of secrets* for RFID tags [9]. Although originally used for a system built around exclusive-OR and a pseudo-random function, it can be applied to other challenge-response building blocks. Damgård and Østergaard Pedersen [5] use the same concept, but speak of *correlated keys.* Nohara et al. in their "K-steps protocol" ([10], also dubbed NIBY) propose to apply trees to the hash-lock setting. They use the term *group IDs* rather than correlated keys, and their trees are unconventional (being of non-uniform depth). Note that all these approaches use a sequence of group- and sub-group IDs to narrow down a tag's identity. As Molnar mentions, *partial keys in such a tree should be chosen independently and uniformly from a key space of sufficient entropy.* Failure to do so would make the system vulnerable to attack. If partial keys are chosen properly, the adversary will have a large key space to search, while the owner of the system can efficiently search through a limited subspace (the actual tree).

The trade-off that exists between efficiency and security in tree-based protocols was already pointed out by Avoine [2], with respect to Molnar's original trees. Because tags share their partial keys, if one tag is compromised (i.e. has its memory probed through invasive tampering), an adversary learns partial keys for several other tags as well. This will enable him to decipher some of their responses, resulting in reduced anonymity and facilitating tracking. Nohl and Evans [11] try to quantify this more precisely. They distinguish between scenarios where compromised tags are chosen in a *selective* or a *random* way, and compute the *information leakage measured in bits.* Their work is one of the few that considers adaptive adversaries (those that selectively choose tags), although not related to the construction of optimal key-trees.

A paper of particular interest is by Buttyán et al. [4], where the concept of trees with *variable branching factors* is introduced, to better preserve anonymity in case of attack. Anonymity in key-trees is expressed in terms of anonymity sets (see Section 2.2). An optimization problem is formulated and solved, and the performance of its solution is evaluated.

In [1], Buttyán et al. attempt to further improve the balance between complexity and privacy in a new "group-based" authentication protocol. However, because the first stage of this protocol includes an encryption of the tag's personal ID, compromise of a group key would result in complete loss of anonymity for all group members. In short, we believe that the results in [1] are flawed, and that merging authentication steps into one step makes for more efficient search, but by definition reduces preservation of anonymity (as follows from results in [4]).

Beye and Veugen [3] also suggest improvements upon [4], by generalizing Buttyán's optimization problem. The resulting trees are provably optimal, and greatly outperform Buttyán trees for some inputs. Beye and Veugen's trees tend to be slightly larger than required, allowing for future system expasion or tag replacement. The results of [4] and [3] are summarized in Section 2.2.

2.1 Notation

This paper bases its notation on that of Buttyán in [4], but makes minor extensions for adaptive adversaries:

- $T = \{t_1, \cdots, t_N\}$: set of all tags in the system
- N: size of T, or actual number of tags in the system
- N': number of leaves in the tree ($\prod(B)$), or maximum number of tags in the system, $N' \geq N$

- c: number of compromised tags
- $P(t_i)$: helper function that returns the anonymity set to which tag t_i belongs
- P_j: anonymity set j, $0 \leq j \leq \ell$
- S: size of a given anonymity set
- $\mathcal{T}$: the set of targeted tags, $\mathcal{T} \subseteq T$
- $\bar{S}(\mathcal{T})$: average size over all anonymity sets for the members of $\mathcal{T}$, in a given configuration
- $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$: $\bar{S}(\mathcal{T})$, averaged over all configurations of c compromised tags across T (Definition 2)
- $\bar{S}_0(c, \mathcal{T})$: lower bound for $\bar{S}(c, \mathcal{T})$, in the worst-case configuration of c compromised tags across $\mathcal{T}$ (see Definition 3)
- $B = (b_1, \ldots, b_d)$: a "branching factor vector" (or tuple), representing a tree of depth d; furthermore, $B \backslash \{b_1, \cdots, b_x\}$ denotes the vector $(b_{x+1}, \ldots, b_d)$
- $R(B)$: resistance to single member compromise for a tree with branching factor vector B. $R(B) \equiv \frac{\bar{S}_{\langle - \rangle}(1,T)}{N} \equiv \frac{\bar{S}_0(1,T)}{N}$
- $R_c(B)$: resistance to c member compromise for a tree with branching factor vector B, $R_c(B) = \bar{S}_{\langle - \rangle}(c, \mathcal{T})/N$
- $\sum(B)$: shorthand for $\sum_{i=1}^{d} b_i$, or the sum over all elements in B
- $\prod(B)$: shorthand for $\prod_{i=1}^{d} b_i$, or the product over all elements in B

Sometimes $\mathcal{T}$ is left out of the notation, e.g. in $\bar{S}(c)$, when $\mathcal{T} = T$. Similarly, c is omitted in case of single member compromise ($c = 1$).

2.2 Key-Trees

Buttyán et al. noted that a time-anonymity trade-off exists, where *narrow, deep trees allow faster search, while wide, shallow trees provide more anonymity.* Obviously, if many tags share the same partial keys, many tags can be excluded from the search space after each authentication stage, implying faster search. The increased anonymity can be intuitively explained by the fact that when partial keys are shared between fewer tags, the amount of information gained by compromising a single tag is limited. Buttyán uses the concept of *anonymity sets* (Pfitzmann and Köhntopp [12], Díaz [6]) to quantify matters.

Definition 1. *Assume a tag t_i sends a given message m (or participates in a protocol execution). For an observer O, the anonymity set $P(t_i)$ contains all tags that O considers possible originators of m. Because all tags in $P(t_i)$ are indistinguishable to O, t_i is anonymous among the other tags in the set.*

Anonymity sets provide a sliding scale for anonymity, where belonging to a larger set implies a greater degree of anonymity. Total anonymity holds if the set encompasses all possible originators in the whole system (one is indistinguishable among all N tags in T), and belonging to a singleton set implies a complete lack of anonymity.

To measure the level of anonymity offered by a tree, the level of anonymity provided to a randomly selected member is used. This *expected size of the anonymity set that a randomly selected member will belong to* is denoted $\bar{S}$ by Buttyán and equals $\bar{S}(c, T)$ in our notation. One could also view it as *the average anonymity set size over all tags*, as shown in Equation 1. Note that $\bar{S}$ can be computed for any given scenario where a tree is broken into anonymity sets.

$$\bar{S} = \sum_{i=1}^{N} \frac{|P(t_i)|}{N} = \sum_{j=1}^{\ell} \frac{|P_j|}{N} |P_j| = \sum_{j=1}^{\ell} \frac{|P_j|^2}{N} \ , \tag{1}$$

where $P(t_i)$ is a function that returns the anonymity set to which tag t_i belongs, P_j denotes an anonymity set and ℓ is the number of sets. Set P_0 is defined as the set containing the compromised tag, e.g. in Figure 1 $P_0 = \{t_{1A\alpha}\}$. The sets P_i, $1 \leq i \leq \ell$, form a partitioning of T.

Buttyán then defines R, the *resistance to single member compromise*, as $\bar{S}$ computed for a scenario where *a single tag* is broken, and then normalizing the result (as in Díaz [6]). Note that because we can freely order the anonymity sets, $c = 1$ leads to a single unique configuration. With its range of $[0, 1]$, R is independent of N, allowing for easy comparison between systems of different sizes.

$$R = \frac{\bar{S}}{N} = \sum_{j=1}^{\ell} \frac{|P_j|^2}{N^2} \ , \tag{2}$$

where P_j denotes an anonymity set, ℓ is the number of sets, d denotes tree depth, and $\bar{S}$ *is computed for the (unique) scenario resulting from single member compromise.* Verify that, in this scenario, the number of sets ℓ is indeed equal to $d + 1$.

Buttyán proposes the use of trees with different, independent branching factors on each level, sorted in descending order (as shown in Figure 1). We will refer to such trees as "*Buttyán trees*", and to trees with a constant branching factor as "*Classic trees*".

Trees will be described by their branching factor vectors $B = (b_1, \ldots, b_d)$, where the variables b_i $(1 \leq i \leq d)$ are positive integers denoting the branching factor at level i.

Buttyán et al. in [4] reach the conclusion that the branching factors near the root contribute more to $\bar{S}$ and R. For trees with variable branching factors this means that a deep, top heavy Buttyán tree can potentially outperform a shallow classic tree.

We rephrase Buttyán et al.'s optimization problem as:

Problem 1. *Given the total number N of members and the upper bound D_{max} on the maximum authentication delay, find the lexicographically largest vector $B = (b_1, \ldots, b_d)$ subject to the following constraints:*

$$\prod(B) = \prod_{i=1}^{d} b_i = N, \text{ and } \sum(B) = \sum_{i=1}^{d} b_i \leq D_{max} \quad . \tag{3}$$

Buttyán et al. provide a *greedy* algorithm that solves this problem recursively. It starts with the prime factorization of N and tries to combine prime factors as long as the sum (authentication time) remains acceptable.

However, Buttyán recognizes that trees need to stand up to more than single tag compromise. Without going into mathematical detail, Buttyán suggests to express $\bar{S}$ for the general case in two different ways:

Definition 2. *$\bar{S}_{\langle - \rangle}(c)$ expresses $\bar{S}(c)$ as the average over all $\binom{N}{c}$ possible distributions of c compromised members across the tag set T.*

Our notation is a natural extension of Buttyán's $\bar{S}_{\langle - \rangle}$, directly incorporating c. Depending on how each successive member is picked from the tree, different anonymity sets are broken down. Buttyán notes that computing $\bar{S}_{\langle - \rangle}$ is hard, and therefore suggests an alternative measure:

Definition 3. *$\bar{S}_0(c)$ represents the* worst-case *value of $\bar{S}(c)$ for all $\binom{N}{c}$ possible distributions of c compromised members across the tag set T.*

Although not stated explicitly in [4], this worst-case value is attained in (any of) the most uniform distributions of c compromised tags across T.

Proof. Assume that we are allowed to choose tags to be compromised sequentially, with the aim to minimize the average anonymity set size. The first compromised tag leads to a unique configuration. Each subsequent compromised tag leads to a new configuration, with more anonymity sets (of varying, decreasing size). To minimize the average set size in the *resulting* configuration, the next tag to be compromised should be chosen from (one of) the largest anonymity set(s) in the *current* configuration. When sorting anonymity sets in ascending order, we observe that this is equivalent to chooseing tags (as) uniformly (as possible given the tree structure) across T. By induction, our claim holds for any c. □

Again, Buttyán's notation $\bar{S}_0$ is generalized to directly incorporate c. Buttyán correctly remarks that $\bar{S}_0(c)$ is far easier to compute, and acts both as a lower bound and an accurate approximation for $\bar{S}_{\langle - \rangle}(c)$.

A different tree construction was proposed by Beye and Veugen [3], who modify Buttyán's optimisation problem to:

Problem 2. *Given the total number N of members and the upper bound D_{max} on the maximum authentication delay, find the vector $B = (b_1, \ldots, b_d)$ that maximizes $R(B)$ subject to the following constraints:*

$$\prod(B) = \prod_{i=1}^{d} b_i \geq N, \text{ and } \sum(B) = \sum_{i=1}^{d} b_i \leq D_{max} \quad . \tag{4}$$

The main idea is that the condition $\prod(B) = N$ is too strict and could lead to inferior solutions. It is shown in [3] how key-trees can be optimized for Problem 2, and that they indeed better retain anonymity when tags are compromised. The number of leaves in the tree, $N' = \prod(B)$, will generally be larger than the actual number N of current tags in the system, and therefore gives an additional buffer of tag IDs which is useful when expanding the system, or replacing compromised tags. Note that because a key-tree only needs to be constructed once (and as a pre-computation stage), the efficiency of the tree-building algorithm is not critical. However, both Buttyán's algorithm, as that of Beye and Veugen are sub-linear in the size of inputs N and D_{max}.

The difference in output can be illustrated with the help of the examples in Table 1:

- Set 1, borrowed from [4], shows that Buttyán's algorithm is not optimal in the setting of Problem 2. The output of Beye and Veugen's algorithm is lexicographically larger, although not much.
- In Set 2, the input contains relatively large primes. Buttyán's algorithm cannot improve upon the Classic tree at all, leaving much room for improvement by Beye and Veugen's algorithm. The difference in performance is about as large as between the Classic and Buttyán trees in Set 1.
- For Set 3, Buttyán's algorithm performs similarly and provides the same output as Beye and Veugen's algorithm. Set 3 is a relatively small example to test whether a large b_d has a positive effect on the entire tree.

Table 1. Test cases

Input	Classic	Buttyán	Beye and Veugen	Hourglass
Set 1: $N = 27000$, $D_{max} = 90$	$(30, 30, 30)$	$(72, 5, 5, 5, 3)$	$(73, 5, 3, 3, 3, 3)$, $N' = 29565$	$(70, 3, 3, 3, 2, 9)$, $N' = 34020$.
Set 2: $N = 24389$, $D_{max} = 100$	$(29, 29, 29)$	$(29, 29, 29)$	$(84, 4, 3, 3, 3, 3)$, $N' = 27216$	$(80, 4, 3, 3, 10)$, $N' = 28800$
Set 3: $N = 1728$, $D_{max} = 36$	$(12, 12, 12)$	$(24, 4, 3, 3, 2)$	$(24, 4, 3, 3, 2)$	$(20, 4, 3, 9)$, $N' = 2160$

3 Adaptive Adversaries

Buttyán et al. in [4] and Beye and Veugen in [3] assume their adversaries to be *non-adaptive* and to select tags at random (naively). Their aim is to provide optimal defense (by maximizing $\bar{S}_{\langle - \rangle}(c, T)$) in the *expected average case* – a uniformly random distribution of compromised tags.

We would like to model other possible lines of attack and see what is required to best preserve anonymity in those cases. First of all, we wish to distinguish the following two goals that an adversary may have:

1. *Universal tracking:* an attacker wants to track *any and all* tags in the system.
2. *Targeted tracking:* an attacker wants to track *certain* tags in the system.

In both scenarios, naive attacks can be mounted by breaking tags at random, thus reducing the expected anonymity set size of the average tag ($\bar{S}_{\langle - \rangle}(c, T)$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$, respectively).

However, clever adversaries may employ additional knowledge to expedite matters. In cryptographic literature, a *side-channel attack* is commonly defined as “any attack based on information gained from the physical implementation of a crypto-system, rather than theoretical weaknesses in the algorithms, which is the aim of cryptanalysis.” The following formal definition is based on that of Köpf and Basin’s [8]:

Definition 4. *Let K be a finite set of secret inputs, M be a finite set of messages, and D be an arbitrary set. We model cryptographic systems as (consisting of) functions of type $F : K \times M \to D$, where we assume that F is invoked by two collaborating callers. One caller is an honest agent that provides a secret argument $k \in K$ and the other caller is a malicious agent (the attacker) that provides the argument $m \in M$. We assume that the attacker has no access to the values of k and $F(k, m)$, but that he can make physical observations about F's implementation I_F that are associated with the computation of $F(k, m)$ (*side-channel information*). The malicious agent performs an attack in order to gather (side-channel) information for deducing k or narrowing down its possible values. Such an attack consists of a* sequence of attack steps*, each with two parts: A query phase in which the attacker decides on a message m and sends it to the system, and a response phase in which he observes I_F while it computes $F(k, m)$.*

In the setting of RFID key-trees, the most obvious example of *side-channel information* is *serialized issuing.* RFID tags are delivered in batches and companies often implement systems in a structured way. Adversaries that are interested in breaking the keys belonging to a particular company, departement or person, will often be able to easily learn some additional information about the RFID tags, and consequently about the construction of the key-tree. Choosing keys from the tree and assigned them to tags in such an orderly fashion can give rise to strong correlations between date of issuing, physical location and key material.

Using this information, an attacker could mount the following attacks:

Ad 1. Universal tracking: to track all tags efficiently, an attacker will aim to make the average anonymity set size over all tags ($\bar{S}$) as small as possible. Assuming that tags are distributed and compromised at random (no known side-channel information can be exploited), the expected remaining anonymity after an attack is equal to $\bar{S}_{\langle - \rangle}(c, T)$ (by definition). In some cases, an unknown order in the tree (i.e. serialized issuing) can work against this adversary’s goals, by making the spread of his compromised tags less uniform than he expects. However, if the adversary manages to exploit such an underlying source of side-channel information, it can help him to select his compromised tags *with a more uniform distribution.* This will shift the results closer to the worst-case value $\bar{S}_0(c, T)$.

Ad 2. Targeted tracking: when attacking a specific subset of tags $\mathcal{T} \subset T$, without side-channel information, the expected result $\bar{S}_{\langle - \rangle}(c, \mathcal{T}) = \bar{S}_{\langle - \rangle}(c, T)$; tags in $\mathcal{T}$ are no different from the average tag. However, if the attacker is able to exploit side-channel information, his efforts can be focussed on breaking tags in $\mathcal{T}$ (or in branches that contain members of $\mathcal{T}$). Note that breaking other tags does have *a limited impact*: it reduces the set size for those tags (if any) in $\mathcal{T}$ which have not had any of their keys revealed yet (and are thus in the same anonymity set). Still, breaking tags in $\mathcal{T}$ itself has by far the largest impact.

Even worse, we argue that a stronger and more readily available source of side-channel information exists, when considering *adaptive attacks*:

Definition 5. *In an* adaptive attack*, the attacker can use the observations made during his first n queries to I_F to choose his message m for the $n+1$st query.*

The most obvious adaptive attack in the current setting would be to test target tags before deciding whether to compromise them or not. Because we already assumed that our attacker has the capability to interrogate a tag and observe its response (for the purpose of tracking), this type of attack would be almost trivial to mount in practice. By simply interrogating a candidate tag, the adversary can determine how many (and even which) keys it shares with his set of "already known keys".

An adaptive adversary has the ability to compromise only those tags that best suit his purposes (i.e. do the most harm with a minimal c), making the following attacks possible:

Ad 1. Universal tracking: if a candidate tag shares too many of its keys with previously compromised tags, it can already be tracked to some extent. It does not form a worthy target for actual compromise, because it would not yield enough new keys. Only tags from unknown parts of the tree, that (mostly) use unknown keys, will be compromised. The resulting distribution is *more uniform than the expected case*, and more closely resembles the fully-uniform worst-case distribution. The rapid breakdown of remaining large anonymity sets will push anonymity metrics towards their worst-case value $\bar{S}_0(c, T)$.
Ad 2. Targeted tracking: if a candidate tag replies with partial keys that are known, it is located in a known part of the tree, and the tag is selected for compromise. This focusses the efforts in a particular sub-tree and rapidly breaks down the anonymity of this subset of tags. Although it would be hard for the attacker (without additional knowledge) to choose *which* part of the tree to attack, a (randomly selected) subset $\mathcal{T}$ can be attacked in particular. Attacks that combine adaptive strategies with other side-channel information (e.g. exploitation of serialized issuing) would have a serious impact the anonymity in *specifically chosen* target sets.

To keep the input to our simulations manageable, we assume that tags in $\mathcal{T}$ are adjacent tags in the tree. We believe this will fit (most) real-world sources of side-channel knowledge. However, to model adversaries trying to track a subset

$\mathcal{T}$ of a different shape (e.g. adaptive testing in a tree with no internal order), a different model would be required.

We generalize $\bar{S}(c,T)$ to represent the anonymity provided to a randomly selected tag $t_i \in \mathcal{T}$, for some target set $\mathcal{T} \subseteq T$.

$$\bar{S}(c,\mathcal{T}) = \sum_{i \in \mathcal{T}} \frac{|P(t_i)|}{|\mathcal{T}|} \ , \tag{5}$$

where $P(t_i)$ is a function that returns the anonymity set to which tag t_i belongs.

Definition 6. *$\bar{S}_{\langle - \rangle}(c,\mathcal{T})$ expresses $\bar{S}(c,\mathcal{T})$ as the* average *over all $\binom{N}{c}$ possible distributions of c compromised members across the tree T.*

Definition 7. *$\bar{S}_0(c,\mathcal{T})$ represents the* worst-case *value of $\bar{S}(c,\mathcal{T})$ for all $\binom{|\mathcal{T}|}{c}$ possible distributions of c compromised members across the (sub-)tree containing $\mathcal{T}$.*

The worst case for tags in $\mathcal{T}$ is attained for those scenarios where all c tags fall into those branches containing members of $\mathcal{T}$, and the spread of these tags is (as close as possible to) uniform. If $c \geq |\mathcal{T}|$, then the remaining tags are spread uniformly (so far as possible) over the remaining branches of the tree.

3.1 Theoretical Impact of Targeted Attacks

Buttyán notes that his result graph for $\bar{S}_0(c,T)/N$ seems to "become a constant" when $c = b_1$. The same trend was observed in the simulation results in [3]. Buttyán mainly uses it to support his claim that the preservation of anonymity relies mostly on the first element of the branching factor vector [4], while we use this observation as our foundation for a better defense against targeted attacks. First we expand upon the informal explanation of this observed behaviour given in [3], which will clearly illustrate the impact of side-channel attacks.

Definition 8. *We define a* turning point *of function $\bar{S}_0(c,T)/N$ as a point where its second derivative exhibits a* jump discontinuity*. In specific, the rate of decline of $\bar{S}_0(c,T)/N$ suddenly slows down by an order of magnitude.*

Corollary 1. *Let c_i be the number of compromised tags for which $\bar{S}_0(c,T)/N$ reaches its i-th* turning point*. Then $c_i = \prod(b_1, b_2, \cdots, b_i)$ (product of the first i branching factors of B). The value of $\bar{S}_0(c_i,T)/N$ will equal $R(B \backslash \{b_1, b_2, \cdots, b_i\})$, in other words is determined only by the remaining branching factors, further down in B.*

Proof. Assume the worst-case scenario, where the distribution of broken tags across T is always at its most uniform (by definition of $\bar{S}_0(c,T)$). This implies that each subsequent tag to be broken, *must come from (one of) the largest remaining anonymity set(s).* For $c \leq b_1$, each newly compromised tag will thus come from a top-level branch containing zero compromised tags. Each compromise reveals one new top-level key, which was previously unknown to the adversary. This key is shared with a whole top-level branch containing $\frac{N}{b_1}$ tags, and its compromise has a large impact on $\bar{S}_0(c, T)/N$.

For $b_1 < c \leq b_1 \cdot b_2$, targets will again fall in the *largest remaining sets*, but these are now housed in the second-level sub-trees and are much smaller than before. All top-level and b_1 of the second-level keys are known, so the following $b_1 \cdot b_2 - 1$ compromised tags each yield one new *second-level* key as the most significant result. These keys are shared among less tags $(\frac{N}{b_1 \cdot b_2})$. Thus, each additional compromise has a smaller impact on $\bar{S}_0(c, T)$. Although $\bar{S}_0(c, T)/N$ does not actually become a constant, the speed of its decline changes drastically.

Such a turning point will occur whenever all keys *from a given level* ℓ have become known to the adversary. There are $\prod(b_1, \cdots, b_\ell)$ such keys, so to reveal them requires (in this worst-case) an equal amount of compromised tags. This means that $c_1 = b_1$, $c_2 = b_1 \cdot b_2$, $\cdots$, $c_d = \prod(B)$. In these cases, all sub-trees (τ_j for $1 \leq j \leq \prod(b_1, \cdots, b_\ell)$) suspended below the branches on level ℓ are identical, and each contains exactly 1 broken tag. From the fact that all tags are housed in an identical subtree, it follows that $\bar{S}_0(c, T)/N$ for the whole tree is equal to the *local* $\bar{S}_0(1, \tau_j)/N'$ for any j. By definition, $\bar{S}_0(1, \tau_j)/N$ (for a tree τ_j containing 1 broken tag) equals $R(B')$ (from Equation 2, also verified by observing a subtree with one compromised member in Figure 1). However, the *local* $\bar{S}_0(1, \tau_j)$ and $R(B')$ are based on τ_j's local $B' = B \backslash \{b_1, b_2, \cdots, b_\ell\}$ and $N' = \prod(b_{\ell+1}, \cdots, b_d)$.

By induction on ℓ, it follows that $\bar{S}_0(c, T)/N$ for the whole tree T assumes the values $R(B \backslash \{b_1\})$ (for $c = c_1$), $R(B \backslash \{b_1, b_2\})$ (for $c = c_2$),$\cdots$,1 (for $c = c_d$) at its turning points. Hence, the remaining anonymity of the remaining tags is dependent only on the remaining branching factors $b_{\ell+1}, \cdots, b_d$ further down in the tree. $\square$

We expect a similar situation to hold for $\bar{S}_{<->}(c, T)$, although we cannot offer a formal description. According to the *Coupon Collector's Problem* [7], one would need to break approximately $b_1 \cdot \log(b_1)$ tags to hit each top-level branch once (assuming branches contain sufficiently many tags, such that breaking tags does not change the probabilities for each branch significantly). However, because tags picked from other branches also (slightly) impact $\bar{S}_{\langle - \rangle}(c, T)$, we expect a *turning trajectory* rather than an exact turning point. Still, we expect the rate of decline for $\bar{S}_{<->}(c, T)$ to depend on the same factors as $\bar{S}_0(c, T)$.

We have seen that given side-channel information, a target subset $\mathcal{T}$ can be rapidly broken down into small anonymity sets. With a Universal Attack based on side-channel information, attackers can cause $\bar{S}_0(c, \mathcal{T})$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ to reach their *turning points* and associated low anonymity values prematurely. Beye & Veugen's Optimized Buttyán trees [3] remain the best defense in this case.

For Targeted Attacks, the situation in the branches containing $\mathcal{T}$ will strongly resemble the one described in the previous paragraphs. Given enough side-channel knowledge, directed attacks inside a smaller sub-tree ignore the top-level branching factor(s). Because the adversary can pick tags from the right branches accurately, the remaining branches offer little to no protection. We therefore postulate that *the values reached by* $\bar{S}_0(c, \mathcal{T})$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ *after the turning points are most important*, not when the turning points are reached.

These values mostly depend on the tail end of B, not the head. Also, *we feel that the difference between belonging to a large and a medium anonymity set is less critical than the difference between belonging to a small anonymity set, and having no anonymity at all.*

3.2 Hourglass Trees

Based on the conclusions of the previous section, we arrive at two conflicting optimization problems. Maximizing the top branching factor is key in defending against Universal and naive attacks, while the lower branching factors play a central role in defending against Targeted Attacks. Without making further assumptions about real-world adversaries, an optimal way of allocating weights cannot be found. To test our hypotheses experimentally, we propose the "*Hourglass*" tree shape. It is top-heavy like Buttyán or Beye & Veugen trees (to provide defense against naive and Universal Attacks), but some weight has been shifted to the *lowest* branching factor to defend against heavy Targeted Attacks. We expect this tree shape to perform better in such scenarios, without sacrificing too much of their strength versus Universal or naive attacks.

Without being able to formulate exact requirements for the tree shape, designing a new tree-building algorithm is not possible. For the purpose of our experiments we will manually adjust B as follows. The bottom branching factor b_d of Beye & Veugen's Optimized Buttyán trees is normally between 2 and 4. We will increase it to a value of around 9, by moving weight from the other b_i, which we expect will provide noticeable results. Note that in some cases, such modification allows for the merging of other branching factors, resulting in a more shallow tree (see Table 1).

4 Simulation

It has already been shown in [3] that Beye & Veugen's trees can yield a lexicographically larger B than Buttyán's approach. We now want to evaluate our Hourglass trees and compare them to Classic and Beye & Veugen trees. To do this, we will compute anonymity measures for each of these tree shapes, under different circumstances.

$\bar{S}_0(c, T)$, $\bar{S}_{\langle - \rangle}(c, T)$ and $\bar{S}_0(c, \mathcal{T})$ will be computed by iterating over all possible scenarios in an efficient way, and taking the (weighted) average and minimum. We will estimate $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ by means of random sampling, for reasons of tractability. Where applicable, anonimity measures for trees with $N' > N$ tags will be scaled by a factor $\frac{N}{N'}$ as discussed in [3].

Side-channel knowledge (or adaptive behavior) is modeled by a probability P for successfully applying knowledge to select a tag from $\mathcal{T}$, where a higher P represents more side-channel knowledge. In case of failure (probability $1 - P$), a random tag is selected from the entire (uncompromised) population. Hence, the total probability of selecting the $(c+1)^{th}$ tag from $\mathcal{T}$ equals $P + \frac{|\mathcal{T}|-c}{N-c}(1 - P)$, excluding the c tags that were previously compromised.

In our experiments, $|\mathcal{T}| = 100$, while $P = 0.1$, 0.5 and 1.0. To approximate $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$, 10,000 random samples were taken and averaged. This resulted in a smooth graph for all inputs, except where $P = 0.1$, for Sets 1 and 2. In these cases 100,000 samples were taken, leading to better results. Running all calculations for $0 \leq c \leq 100$ was still feasible on the hardware used (Pentium-IV 2.0GHz running Windows XP).

Table 1 shows the three input sets for which we have evaluated the Classic, Beye & Veugen and Hourglass trees.

4.1 Graphs for Naive Attacks

Figures 2, 3 and 4 show the performance of the different trees, in the case of naive attacks (compromise at random, without side-channel knowledge). The datasets are selected by relevance, and we discuss how these results relate to our hypotheses and claims.

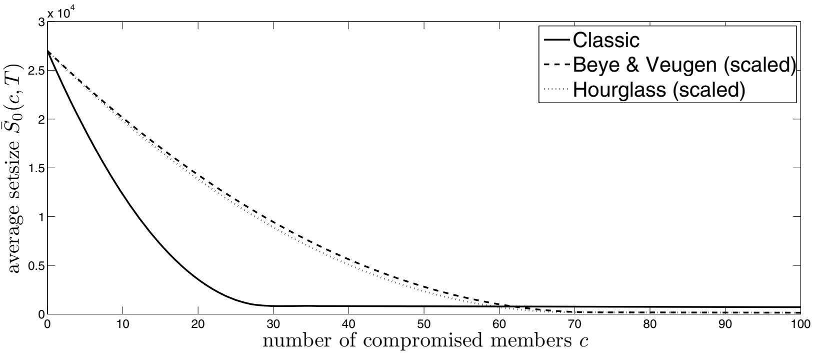

Fig. 2. $\bar{S}_0(c, T)$ for Set 1

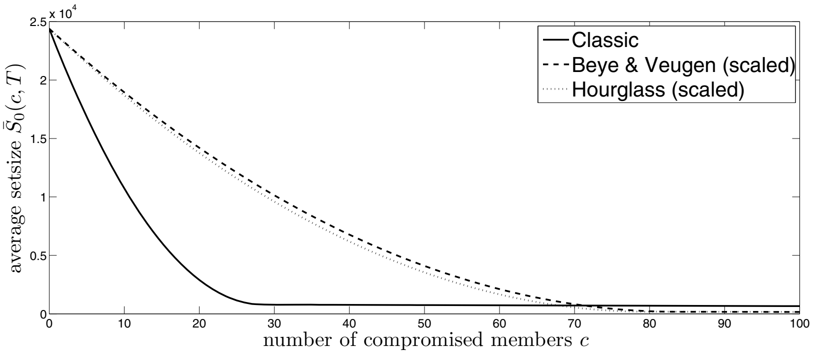

Fig. 3. $\bar{S}_0(c, T)$ for Set 2

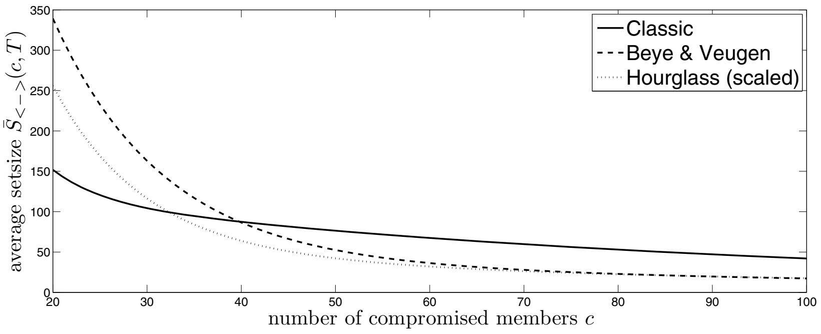

Fig. 4. $\bar{S}_{\langle - \rangle}(c, T)$ for Set 3

Figure 2 shows the Beye & Veugen and Hourglass trees performing similarly for Set 1, in terms of $\bar{S}_0(c, T)$. The same trend was observed for Set 2 (Figure 3), and for $S_{\langle - \rangle}(c, T)$ in Set 3 (Figure 4). The fact that *the differences between Hourglass and Beye & Veugen trees is not large in the absence of side-channel knowledge*, again supports our claim that the value of $\bar{S}$ depends mostly on the first element of B. It also confirms our hypothesis that a small decrease in b_1 does not have major negative impact on the anonymity in case of naive attacks.

The Hourglass tree shape seems to offer no benefit in the non-adaptive (naive) scenario's (as expected), and performs only slightly worse than the other trees in terms of $S_{\langle - \rangle}(c, T)$ (as in Figure 4).

4.2 Graphs for Targeted Attacks

Figures 5 and 6 show the results in case of Targeted Attacks (on a target subset $\mathcal{T}$ of size 100, with the aid of side-channel knowledge or adaptive testing). Again, a selection of result datasets is shown, based on relevance.

It is interesting to observe Classic trees performing very well in these scenario's, which is due to their large value of b_d. As expected, superior results for $\bar{S}_0(c, \mathcal{T})$ are attained with Hourglass trees, second only to Classic Trees. They outperform Beye & Veugen's Optimized Buttyán trees significantly (Figure 6). However, Beye & Veugen's trees can perform better in terms of $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ at low c values, as was the case for Set 3 ($0 \leq c \leq 20$) in Figure 5.

For $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$, Hourglass trees under perform in scenarios with low side-channel knowledge ($P = 0.1$). Although we did not expect this, it can be explained by the fact that the expected average distribution will remain closer to uniform than in cases with more side-channel knowledge – in other words, we remain close to a naive attack. For low P values, $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ behaves much like $\bar{S}_{\langle - \rangle}(c, T)$, for which we have seen that Hourglass trees degrade performance (slightly).

In case of higher side-channel knowledge, the strength of Hourglass trees becomes more apparent. An intersection point exists (see Figure 5), where

Hourglass trees start outperforming Beye & Veugen trees. This point arises earlier when stronger side-channel knowledge is available. Indeed, for the worst-case $\bar{S}_0(c, \mathcal{T}), P = 1.0$, the turning point comes very early ($c = 20$), and the Hourglass tree performs significantly better than its competitor (see Figure 5).

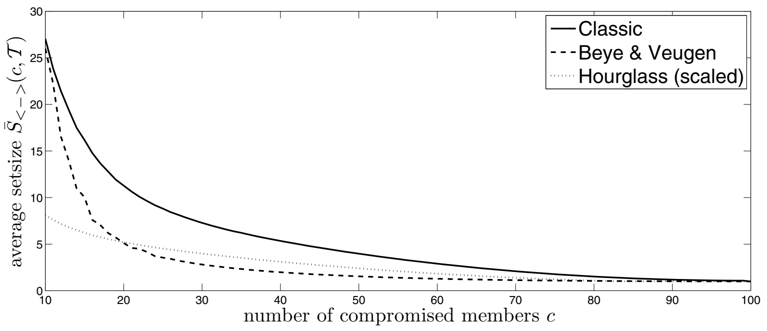

Fig. 5. $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ for Set 3, $P = 1.0$

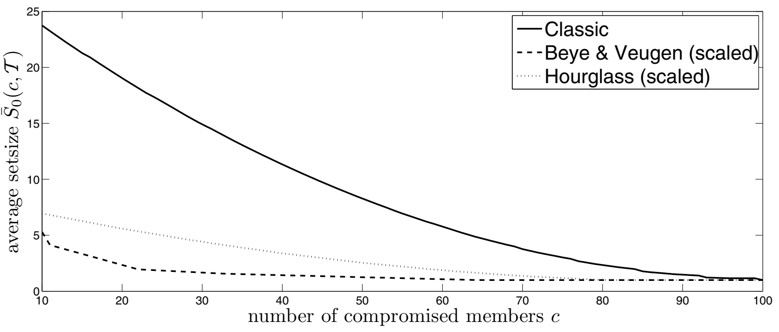

Fig. 6. $\bar{S}_0(c, \mathcal{T})$ for Set 2

5 Conclusions and Future Work

Simulation results support our intuition with regards to our proposed anonymity measures $\bar{S}_0(c, \mathcal{T})$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$. They represent the anonymity in a target subset $\mathcal{T}$ in the same way that $\bar{S}_0(c, T)$ and $\bar{S}_{\langle - \rangle}(c, T)$ do for the whole tree T. Their rate of decline is directly related to the branching factors. As anticipated, the remaining anonymity in case of Targeted Attacks depends heavily on the branching factors located in the tail of B. This means that maximizing $\bar{S}(c, T)$ and $\bar{S}(c, \mathcal{T})$ are indeed contradicting goals, and real-world assumptions regarding attackers will dictate where the emphasis should lie.

The Beye & Veugen trees perform well in terms of $\bar{S}_0(c, T)$ and $\bar{S}_{\langle - \rangle}(c, T)$, but not for $\bar{S}_0(c, \mathcal{T})$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$ with high side-channel knowledge and c values, as we expected.

The proposed *Hourglass* trees perform best in terms of $\bar{S}_0(c, \mathcal{T})$ and $\bar{S}_{\langle - \rangle}(c, \mathcal{T})$, but only with high side-channel knowledge and c values. Their performance in terms of $\bar{S}_0(c, T)$ and $\bar{S}_{\langle - \rangle}(c, T)$ is only slightly below that of Beye & Veugen trees. To summarize: if we expect heavy Targeted Attacks, Hourglass trees will provide prolonged protection, at only a small "cost" in overall anonymity in other attack scenarios.

Some possible directions for future work are:

- Better simulation of real-world scenarios, specifically side-channel knowledge and adversarial behavior. For example modeling non-continuous target sets, and realistically estimating the size of target sets, minimum and maximum values for c, and the amount and nature of side-channel knowledge available to adversaries.
- Given the trade-off between maximizing $\bar{S}(c, T)$ and $\bar{S}(c, \mathcal{T})$, find a way to prioritise between defending against targeted and general attacks, and design an algorithm to optimize trees accordingly.
- Look into new measures for anonymity which do not show absolute declines, but the ratio between current anonymity set size and the decline caused by the next tag being compromised. This would fit the idea that a decline in set size from 1,000 to 999 does not have the same impact as going from a set of size 2 to having no anonymity at all.

References

1. Avoine, G., Buttyán, L., Holczer, T., Vajda, I.: Group-based private authentication. In: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–6 (2007)
2. Avoine, G., Dysli, E., Oechslin, P.: Reducing Time Complexity in RFID Systems. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 291–306. Springer, Heidelberg (2006)
3. Beye, M., Veugen, T.: Improved Anonymity for Key-trees. Cryptology ePrint Archive (2011)
4. Buttyán, L., Holczer, T., Vajda, I.: Optimal Key-Trees for Tree-Based Private Authentication. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 332–350. Springer, Heidelberg (2006)
5. Damgård, I., Pedersen, M.Ø.: RFID Security: Tradeoffs between Security and Efficiency. Cryptology ePrint Archive, Report 2006/234 (2006)
6. Díaz, C.: Anonymity Metrics Revisited. In: Dolev, S., Ostrovsky, R., Pfitzmann, A. (eds.) Anonymous Communication and its Applications. Dagstuhl Seminar Proceedings, vol. 05411, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl (2006)
7. Flajolet, P., Gardy, D., Thimonier, L.: Birthday paradox, coupon collectors, caching algorithms and self-organizing search. Discrete Appl. Math. 39(3), 207–229 (1992)

8. Köpf, B., Basin, D.A.: An information-theoretic model for adaptive side-channel attacks. In: ACM Conference on Computer and Communications Security, pp. 286–296 (2007)
9. Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures. In: CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 210–219. ACM, New York (2004)
10. Nohara, Y., Nakamura, T., Baba, K., Inoue, S., Yasuura, H.: Unlinkable identification for large-scale rfid systems. Information and Media Technologies 1(2), 1182–1190 (2006)
11. Nohl, K., Evans, D.: Quantifying Information Leakage in Tree-Based Hash Protocols (Short Paper). In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 228–237. Springer, Heidelberg (2006)
12. Pfitzmann, A., Köhntopp, M.: Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 1–9. Springer, Heidelberg (2001)