Does Counting Still Count? Revisiting the Security of Counting based User Authentication Protocols against Statistical Attacks [*]

Hassan Jameel Asghar
Centre for Advanced Computing
Algorithms and Cryptography
Department of Computing, Faculty of Science
Macquarie University, Sydney, Australia
hassan.asghar@mq.edu.au

Shujun Li
Department of Computing
Faculty of Engineering and Physical Sciences
University of Surrey, Guildford, Surrey, UK
shujun.li@surrey.ac.uk

Ron Steinfeld
Clayton School of Information Technology
Faculty of Information Technology
Monash University, Clayton, Australia
Email: ron.steinfeld@monash.edu

Josef Pieprzyk
Centre for Advanced Computing
Algorithms and Cryptography
Department of Computing, Faculty of Science
Macquarie University, Sydney, Australia
josef.pieprzyk@mq.edu.au

Abstract

At NDSS 2012, Yan et al. analyzed the security of several challenge-response type user authentication protocols against passive observers, and proposed a generic counting based statistical attack to recover the secret of some counting based protocols given a number of observed authentication sessions. Roughly speaking, the attack is based on the fact that secret (pass) objects appear in challenges with a different probability from non-secret (decoy) objects when the responses are taken into account. Although they mentioned that a protocol susceptible to this attack should minimize this difference, they did not give details as to how this can be achieved barring a few suggestions. In this paper, we attempt to fill this gap by generalizing the attack with a much more comprehensive theoretical analysis. Our treatment is more quantitative which enables us to describe a method to theoretically estimate a lower bound on the number of sessions a protocol can be safely used against the attack. Our results include 1) two proposed fixes to make counting protocols practically safe against the attack at the cost of usability, 2) the observation that the attack can be used on non-counting based protocols too as long as challenge generation is contrived, 3) and two main design principles for user authentication protocols which can be considered as extensions of the principles from Yan et al. This detailed theoretical treatment can be used as a guideline during the design of counting based protocols to determine their susceptibility to this attack. The Foxtail protocol, one of the protocols analyzed by Yan et al., is used as a representative to illustrate our theoretical and experimental results.

1. Introduction

Every now and then we are informed via the media about people's credentials being stolen and misused through new ways. As technology thrives to provide convenience to users through ubiquitous access to their assets, attackers are likewise adapting by developing new ways of impersonating the user. Prime examples of such methods include credit card skimmers, hidden cameras over Automated Teller Machines (ATMs), shoulder-surfing, malicious computer terminals in public spaces, and phishing attacks spoofing users to disclose their credentials. By far the most prevalent means of establishing a user's identity is through password or Personal Identification Number (PIN). It is straightforward to see that this method of authentication is not secure under the aforementioned threats, as the password or PIN can be obtained by an attacker through a single observation. Just recently, in the state of Louisiana in the United States, 25 cases of debit card skimmers have been reported which have cost the victims each between 500 to 600 dollars [21].

[*] This paper was published in *Proceedings of 20th Network & Distributed System Security Symposium* (NDSS 2013), February 24-27, 2013, San Diego, CA, USA, by Internet Society. The full edition of this paper is available at http://eprint.iacr.org/2010/478.

These devices record users' PINs which can then be used to withdraw money through ATMs using a clone of the debit card. The problem can be mitigated to some extent by using more sophisticated methods of authentication, for instance, biometric-based authentication or by using a trusted device. However, these methods introduce vulnerabilities of their own such as potential tampering with the biometric-sensing equipment or the theft of the trusted device.

Generally speaking, in a user authentication system an observer is an attacker who can observe and record one or more login (authentication) sessions between the user $\mathcal{U}$ (prover) and the server $\mathcal{S}$ (verifier) in order to obtain the user's password P. The abstract entity, observer, can be realized by a human via shoulder-surfing or by a machine such as a hidden camera, or a combination of both. A successful attack enables the observer to impersonate the user by deceiving the server into falsely believing that he is the legitimate user. A stronger notion is that of an active attacker who, in addition to having the capability of performing an observer attack, can also maliciously manipulate the communication between the user and the server to gain further advantage in impersonating the user. One solution to the observer problem is challenge-response type authentication, which has the following general structure:

```
1: $\mathcal{U}$ sends his ID to $\mathcal{S}$.
2: for $i = 1$ to $r$ do
3:     $\mathcal{S}$ sends the challenge $\mathcal{C}_i$ to $\mathcal{U}$.
4:     $\mathcal{U}$ sends the response $\mathcal{R}_i$ to $\mathcal{S}$.
5: if the responses are correct for all the $r$ rounds then $\mathcal{S}$
   accepts $\mathcal{U}$ else $\mathcal{S}$ rejects $\mathcal{U}$.
```

Since $\mathcal{S}$ shares the password with $\mathcal{U}$, it can verify each response by comparing it with its own computation. The challenge-response pairs $\{(\mathcal{C}_i, \mathcal{R}_i)\}_{i=1}^{r}$ are communicated in the open. The user responds by computing a publicly known function f of the password and the challenge. The function should be such that the password cannot be revealed upon the observation of a sufficiently large number of challenge-response pairs. The first known effort to build a challenge-response protocol secure under the threat of an observer was made by Matsumoto and Imai [16] in 1991, but a solution with both high security and acceptable usability remains an open problem to date [13, 27].

Challenge-response protocols often follow the k-out-of-n paradigm in which the password is a k-element subset of a set of n objects. The objects forming the password constitute the pass-objects whereas the remaining $n - k$ objects are the decoy objects. One type of challenge-response protocols is counting based protocols in which the user counts the number of pass-objects present in a challenge and then responds by computing a function f of the count. Normally, the number of objects present in a challenge is less than n and is called the *window* size. The appeal of counting based protocols over others is their apparent usability advantage since the user has to merely recognize and count the number of objects present in a challenge. In [27, § 4.4], Yan et al. introduced a passive attack, called counting based statistical analysis, which reveals the password in counting based protocols after the observation of far fewer sessions than original claims. Briefly, the attack involves observation of a sufficiently large number of challenge-response pairs from a protocol followed by an analysis of the difference in occurrence of δ-tuples of pass and decoy objects in the challenges corresponding to different responses. Here δ is any number between 1 and k inclusive [27]. Yan et al. did not give a theoretical analysis as to why their attack works apart from a demonstration of the attack on the Foxtail protocol for the case $\delta = 2$, i.e., by considering pairs of objects. The Foxtail protocol was proposed in [13] as a counting based protocol secure against observation.

OUR CONTRIBUTIONS. In this paper, we revisit the reason why existing counting based protocols are not secure against the attack from Yan et al. We point out that their attack on Foxtail is actually based on a simpler attack that analyzes individual objects instead of pairs. In [27], Yan et al. stated that the simpler attack is not possible due to the inclusion of a condition in Foxtail to ensure the number of pass and decoy objects are the same in each challenge. This implies that the attack is applicable to all counting based protocols which include a similar condition. We put the attack from Yan et al. into context by observing that in contrast with other statistical attacks, this attack merely analyzes the occurrence of tuples of pass and decoy objects given challenge-response pairs. To this end, we call the attack *frequency analysis*, to borrow a term from classical cryptanalysis. We report a theoretical study on the reason why frequency analysis works on protocols that have contrived challenges [27, §4.2]. We also show that the attack is applicable to some non-counting based protocols. Where possible, we give quantitative details of the general attack. In addition, we demonstrate a method to theoretically estimate a lower bound on the number of sessions required to distinguish a pass-object tuple from its decoy counterpart. This serves as a criterion to decide whether the protocol in question is susceptible to frequency analysis in a practical setting. We then discuss what is required to make a protocol practically safe from this attack, introducing two new principles as a result. Based on the proposed new principles, we suggest two fixes against the attack; one general to all counting based protocols and the other specific to Foxtail. We show how these fixes render the protocols practically safe from frequency analysis, although they do incur extra usability costs. We demonstrate these amendments using Foxtail as an example. To the best of our knowledge, the resulting protocols are by far the most usable protocols which remain practically secure against frequency analysis. We invite the community to scrutinize the security of our protocols and to investigate if there is a more usable and secure (against frequency analysis) counting based protocol than our proposals.

ORGANIZATION. The rest of the paper is organized as follows. We discuss the related work in Section 2. Section 3 describes a general model of counting based protocols using the Foxtail protocol as an example to concretize the attack from Yan et al. A detailed theoretical explanation of why their attack works is given in Section 4. We give a detailed theoretical treatment of the generalized attack in Section 5, and discuss the impact of different ways of generating challenges in counting based protocols in Section 6. Section 7 discusses the applications of the attack to non-counting based protocols. We introduce two new principles and fixes against the attack in Section 8. Section 9 contains concluding remarks.

2. Related Work

In accordance with the theme of the paper, we shall restrict this brief review of literature to generic attacks and design principles related to challenge-response type user authentication protocols. Furthermore, since we consider passive adversaries only, we shall not discuss active attacks. As mentioned before, the first attempt at constructing a protocol secure against a powerful passive adversary dates back to Matsumoto and Imai [16]. The protocol was shown to be susceptible to a passive attack in [23], but the attack is specific to the protocol. Li and Teng [14] proposed another attack on the same protocol which can be categorized as an intersection attack. Roughly speaking, an intersection attack attempts to find the password by checking if a set of candidates for the password satisfy a specific criterion. As more and more sessions are observed, the size of the candidate set diminishes as members failing the criterion are discarded. A brute force attack can be considered as the most computationally intensive intersection attack. Intersection attack has also been demonstrated on Predicated-based Authentication Service (PAS) [26] by Li et al. in [11]. In [15], Matsumoto proposed protocols that are secure up until a certain number of sessions after which Gaussian elimination can be used to find the password. This algebraic attack can find the password by observing a number of observed challenge-response pairs which is linear in the size of the challenge. Subsequent protocols that have incorporated resilience against Gaussian elimination in their design include Hopper and Blum (HB) [8], Foxtail [13] and Asghar, Pieprzyk and Wang (APW) [2] protocols. Another algebraic attack was demonstrated on the virtual password scheme [9, 10] by Li et al. in [12]. Satisfiability (SAT) solvers have been used by Golle and Wagner [7] to demonstrate the insecurity of Weinshall's Cognitive Authentication Scheme (CAS) [24]. SAT solvers can find the secret once the information from the challenge-response pairs can be translated in the form of SAT clauses.

Hopper and Blum [8] introduced a meet-in-the-middle attack of time complexity $\mathcal{O}(\binom{n}{k/2})$ which is much more efficient than the brute force attack. They also showed that it is not possible to circumvent this attack even by introducing noise, implying that the attack is also applicable to the HB protocol introduced in the same paper. Yan et al. also showed that the introduction of noise does not preclude brute-force attack [27, §4.4]. An improved meet-in-the-middle attack was shown by Asghar et al. in [2] with a time-complexity of $\mathcal{O}(\binom{n/2}{k/2})$. In [3], it is shown that the other protocol from [8], namely the *sum of k mins* protocol, is based on a fixed-parameter intractable problem, which roughly means that the best possible worst-case algorithms to find the secret will take time $n^{f(k)}$ for some function f. Li and Shum introduced several design principles to overcome some of the vulnerabilities in the previous protocols [13]. The principle relevant to security against passive attacks suggests including uncertainty in the protocol; for instance, by occasionally sending an intentionally wrong response. Coskun and Herley introduced a generic attack on protocols which use a subset of the password to compute a response in each round [6]. The attack has a far lower time-complexity than the brute force attack if the subset of the password used is small. The attack is based on the observation that candidates for the password which are different from the password in a small number of objects can be easily distinguished from others since their responses are much more similar to those computed using the password.

Yan et al. specified several design principles in [27]. The first of them recommends a large secret space, and it even applies to protocols that have hidden responses such as the Undercover protocol from [19]. This vulnerability on Undercover was also independently demonstrated by Perković et al. with the same intersection attack [17]. Note that if the challenges are fixed instead of randomized, Undercover is not vulnerable to this attack [17, §4.3.1]. The second principle suggests that the subset of password used for each round of a challenge-response protocol should be large. The third principle relates to the distribution of objects in a challenge when a certain constraint on the number of secret-objects present in a challenge is required, e.g., at least 3 secret objects be present in the challenge as in the case of the Convex Hull Click (CHC) protocol [20, 25]. The principle suggests that in such a construction, all objects should be present in a challenge. The fourth principle is related to protocols in which it is possible to construct a *probabilistic decision tree* based on the challenge-response pairs. To avoid an attack based on this, it should not be possible to distinguish individual objects in a decision path consistent with the observed challenge-response pair. According to the authors, this attack is specially dangerous on protocols which have any weight or orientation information associated with the objects such as CAS [24]. The last principle states that the protocols be designed such that they are resistant to counting based statistical analysis, which is the focus of this paper.

3. Counting based Protocols and Foxtail

Although the attack proposed by Yan et al. in [13] was demonstrated on the Foxtail protocol[1] only, it can in general be applied to a class of protocols that are based on counting pass-objects in a challenge. It is therefore essential to present a general model of counting based protocols.

3.1. General Model of Counting based Protocols

Counting based protocols are multi-round challenge-response protocols based on a pool (set) of n objects $O = \{o_1, \dots, o_n\}$. The password, challenges and responses are defined as follows:

- The password P is a k-element subset of O, where $k < n$. We shall call an element of P, a pass-object. P can be represented as the set $\{\rho_1, \rho_2, \dots, \rho_k\}$, where ρ_i are the pass-objects.[2] The set $D = O - P$ is called the decoy set, and all its elements are called decoy objects. A decoy object shall be denoted by d. Notice that since is O fixed, D is also fixed.

- Each challenge $\mathcal{C}_i$ is a *finite sequence* of objects from a *subset* of O. A sequence is different from a set in that each element can occur multiple times and the order of the elements matters. The notation $\#\mathcal{C}_i$ denotes the length of the sequence. For any subset $O' \subseteq O$, let $\mathcal{C}_i(O')$ be the subsequence of $\mathcal{C}_i$ containing objects from O'. $\mathcal{C}_i(O')$ is empty if no object from O' is in $\mathcal{C}_i$. $\#\mathcal{C}_i(O')$ denotes the length of this subsequence. From these definitions, we see that for the password P,

$$\#\mathcal{C}_i(P) = \sum_{\rho \in P} \#\mathcal{C}_i(\{\rho\}).$$

  Informally, $\#\mathcal{C}_i(P)$ is the sum of all occurrences of pass-objects in the challenge $\mathcal{C}_i$.

- The response is defined as $\mathcal{R}_i = f(\#\mathcal{C}_i(P))$, where f is a function, called the response function. In the simplest case, $\mathcal{R}_i = \#\mathcal{C}_i(P)$, thus making a response only requires counting the occurrence of pass-objects in the challenge. Another example of f is $\bmod q$ for any $q \geq 2$.

3.2. Foxtail and the Attack from Yan et al.

Since Yan et al. used the Foxtail protocol to demonstrate their attack, we describe the protocol here for reference. It should be noted that the attack can be generalized for any counting based protocol. The Foxtail protocol proposed in [13] is a counting based protocol which implements the challenge and the response function f as follows:

- A challenge $\mathcal{C}_i$ contains two *half-challenges*, each composed of l objects.
  - The first half $\mathcal{C}_{i,1}$ is generated following the so-called *Uni-rule* (meaning uniform rule) such that $\mathcal{C}_{i,1}$ contains 0, 1, 2 or 3 pass-objects with equal probability. This implies that the number of pass-objects in the first half challenge cannot be more than 3.
  - The second half $\mathcal{C}_{i,2}$ is generated through the so-called *Rand-rule* (meaning random rule) which means that each element in $\mathcal{C}_{i,2}$ is generated by randomly picking an object from the object pool O.
- The response function is as follows:

$$\begin{aligned}\mathcal{R}_i = f(\#\mathcal{C}_i(P)) &= \left\lfloor \frac{\#\mathcal{C}_i(P) \bmod 4}{2} \right\rfloor \\ &= \left\lfloor \frac{(\#\mathcal{C}_{i,1}(P) + \#\mathcal{C}_{i,2}(P)) \bmod 4}{2} \right\rfloor \in \{0, 1\}.\end{aligned}$$

The objects from the two half-challenges are randomly shuffled in $\mathcal{C}_i$ so that the attacker does not know which object belongs to which of the two half-challenges. To ensure that the pass and decoy objects occur in the challenge with the same probability, $3n = 2kl$ must hold [13]. A recommended set of parameter values for Foxtail is $(n, k, l) = (140, 14, 15)$. This setting was believed to be able to offer a balance between usability and security against all known attacks [13]. The number of rounds r is 20 to make sure that the probability of randomly guessing the response to a challenge is approximately one in a million. Unless otherwise mentioned, r is understood to be 20.

In [27], Yan et al. pointed out that the above mentioned Foxtail protocol is not secure against a 2-dimensional counting based attack which exploits the following observation about Foxtail with the recommended parameter values:

[1] Yan et al. call this protocol SecHCI, but SecHCI is actually used as an umbrella term in [13] for protocols intended to be secure against observer attacks. We therefore use the term Foxtail instead, which alludes to the specific function described above.

[2] The reason for using ρ instead of p for pass-objects is that the latter symbol is reserved for probabilities.

- In all challenges corresponding to a 0-response, there are 0.599 more pairs among pass-objects on average.
- In all challenges corresponding to a 1-response, there are 0.599 fewer pairs among pass-objects on average.

Based on this *response-dependent* imbalance, a score can be maintained for each object pair in two tables, one for each response. That is, for each challenge-response pair we see if the object-pair is present or not. If it is present, we increment the score by 1 in the table corresponding to the response. The final score is the difference in scores between the 0-response and the 1-response. By observing a sufficient number of authentication sessions, it is possible to differentiate pass-object pairs from decoy-object pairs because the former tend to have higher scores. Yan et al. call this attack *2-dimensional counting attack* and attributed its success to the correlation between the password and the responses in two-dimensional space.

In this paper, we shall refer to this type of attack as *response-dependent frequency analysis*, abbreviated as RDFA. The name alludes to the observation that this type of attack essentially analyzes the difference in the frequency of occurrence of objects in challenges. A *response-independent* frequency analysis is also possible as noted by Yan et al. [27]. In the Foxtail protocol, the condition $3n = 2kl$ is required to prevent precisely this attack. Again, for brevity's sake, we shall refer to this variant as RIFA. More generally, we will refer to a RDFA (resp., RIFA) involving δ-tuples of objects as δ-dimensional RDFA or δD-RDFA (resp., δD-RIFA). The umbrella term, frequency analysis, covers both RIFA and RDFA.

4. Revisiting the Attack from Yan et al. on Foxtail

While Yan et al. attributed the success of 2D-RDFA to the correlation in 2-dimensional (2D) space, they did not give a theoretical explanation as to why their attack works other than demonstrating the attack with experimental results. In this section we show that a simpler attack also works in 1-dimensional (1D) space, leading to the finding that the attack from Yan et al. is in fact a generalization of the 1D attack to 2D space. In other words, we show that 1D-RDFA is also possible on the Foxtail protocol, and it is the theoretical basis of 2D-RDFA proposed by Yan et al. Note that they mentioned that due to the condition $3n = 2kl$, 1D frequency analysis is not possible on Foxtail [27, §4.4]. However, as we will show later this condition only precludes 1D-RIFA. Leaving general results aside for now, we begin with the detailed description of 1D-RDFA and how it works on Foxtail.

In the original security analysis of the Foxtail protocol given in [13], the occurrence of different objects in the challenges are calculated without considering the responses. However, since the passive observer can see the responses, he can divide the challenges into two sets, each corresponding to one response bit, and then analyze them separately. To proceed with this idea, let us see what are the values of $c_1 = \#\mathcal{C}_{i,1}(P)$ and $c_2 = \#\mathcal{C}_{i,2}(P)$ when the response is known. We can see, for instance, that when $\mathcal{R}_i = 0$, (c_1, c_2) can take the values

$$(0,0),(0,1),(1,0),(1,3),(2,2),(2,3),(3,1),(3,2),\ldots,$$

and when $\mathcal{R}_i = 1$, (c_1, c_2) can have the values

$$(0,2),(0,3),(1,1),(1,2),(2,0),(2,1),(3,0),(3,3),\ldots.$$

Observing the above, we can see that the expected occurrence of any pass-object ρ could be different under the two conditions. In addition, the expected occurrence of any pass-object could also be different from that of any decoy object. To find an explicit equation for the expected occurrence of any object, we need to define a few notations for succinctness. Let p'_{c_1} denote the probability of the event $\#\mathcal{C}_{i,1}(P) = c_1$, and let p_{c_2} denote the probability of the event $\#\mathcal{C}_{i,2}(P) = c_2$. Further, denote the probability that ρ appears when a half-challenge contains c pass-objects by q_c.[3] Assume $\mathcal{R} \in \{0,1\}$. Define the sets

$$F_{\mathcal{R}}(c_1) = \left\{ c_2 \middle| c_2 \in \{0,\ldots,\min(k,l)\} \text{ and } \left\lfloor \frac{(c_1+c_2) \bmod 4}{2} \right\rfloor = \mathcal{R} \right\}. \quad (1)$$

Note that there can be a maximum of $\min(k,l)$ pass-objects in the second half-challenge, therefore c_2 is always less than or equal to this number. On the other hand, the first half-challenge can have a maximum of 3 pass-objects. For compactness, we shall use $F_{\mathcal{R}}$ to denote $F_{\mathcal{R}}(c_1)$. The choice of the symbol F for these sets alludes to the fact that it is dependent on the response function f. Define

$$\tilde{p}_{\mathcal{R}} = \sum_{c_1=0}^{3} \sum_{c_2 \in F_{\mathcal{R}}} p'_{c_1} p_{c_2}, \quad (2)$$

which denotes the probability that the response is $\mathcal{R}$.

Now, denote the expected occurrence of a pass-object in a challenge corresponding to response $\mathcal{R}$ by $\xi_\rho(n,k,l,\mathcal{R})$ or $\xi_\rho(\mathcal{R})$ for simplicity. Then

$$\begin{aligned}\xi_\rho(\mathcal{R}) &= \sum_{c_1=0}^{3} \sum_{c_2 \in F_{\mathcal{R}}} \frac{p'_{c_1} p_{c_2}}{\tilde{p}_{\mathcal{R}}} (q_{c_1} + q_{c_2}) \\ &= \frac{1}{\tilde{p}_{\mathcal{R}}} \sum_{c_1=0}^{3} \sum_{c_2 \in F_{\mathcal{R}}} p'_{c_1} p_{c_2} (q_{c_1} + q_{c_2}). \quad (3)\end{aligned}$$

[3] Note that q_c is the same for both half-challenges. For a fixed c, there is no distinction as to which pass-objects are chosen.

For the Foxtail protocol under study, the three probabilities have the following values:

$$p'_{c_1} = \frac{1}{4},\ p_{c_2} = \frac{\binom{k}{c_2}\binom{n-k}{l-c_2}}{\binom{n}{l}},\ q_c = \frac{c}{k}. \qquad (4)$$

Similarly, we can derive the expected occurrence of a decoy object d per challenge for both responses. Denote this by $\xi_d(\mathcal{R})$. We have

$$\xi_d(\mathcal{R}) = \frac{1}{\tilde{p}_{\mathcal{R}}} \sum_{c_1=0}^{3} \sum_{c_2 \in F_{\mathcal{R}}} p'_{c_1} p_{c_2} (q'_{c_1} + q'_{c_2}). \qquad (5)$$

Here, $q'_c = \frac{l-c}{n-k}$ denotes the probability that d appears when a half-challenge contains c pass-objects. From these equations, we see that it is possible to differentiate a pass-object from a decoy object if any of the three inequalities holds:

- $\xi_\rho(0) \neq \xi_d(0)$.
- $\xi_\rho(1) \neq \xi_d(1)$.
- $\xi_\rho(0) - \xi_\rho(1) \neq \xi_d(0) - \xi_d(1)$.

For the recommended parameters of Foxtail, we estimated the values of $\xi_\rho(0)$, $\xi_\rho(1)$, $\xi_d(0)$ and $\xi_d(1)$ using Monte Carlo simulations. The results are as follows: $\xi_\rho(0) \approx 0.218548$, $\xi_\rho(1) \approx 0.210024$, $\xi_d(0) \approx 0.213812$, $\xi_d(1) \approx 0.214759$. One can see that the first two inequalities hold. It is obvious that the third inequality also holds: $\xi_\rho(0) - \xi_\rho(1) \approx 0.008524 > \xi_d(0) - \xi_d(1) \approx -0.000947$. Not only does $\xi_\rho(0) - \xi_\rho(1)$ has a different sign from $\xi_d(0) - \xi_d(1)$, but also its amplitude is 9 times higher than the latter. These results merely show that the inequalities hold when the parameters are $(n, k, l) = (140, 14, 15)$. It still leaves open the possibility that there is a set of values for which these inequalities do not hold. Unfortunately, in the following, we show that this is not possible if $3n = 2kl$.

Theorem 1. *Let $\xi_\rho(\mathcal{R})$ and $\xi_d(\mathcal{R})$ be as defined in Eqs. (3) and (5), then*

(1) $\xi_\rho(0) - \xi_\rho(1) = -\frac{n-k}{k}(\xi_d(0) - \xi_d(1))$.

(2) If $3n = 2kl$, *then* $\xi_\rho(0) + \xi_\rho(1) = \xi_d(0) + \xi_d(1) = \frac{4l}{n}$.

Proof. The rather tedious proof of this theorem is given in the full edition of this paper. □

Corollary 1. *Let $\xi_\rho(\mathcal{R})$ and $\xi_d(\mathcal{R})$ be as defined in Eqs. (3) and (5), then*

(1) $\xi_\rho(0) = \xi_\rho(1)$ *if and only if* $\xi_\rho(0) - \xi_\rho(1) = \xi_d(0) - \xi_d(1)$.

Further if $3n = 2kl$, *then*

(2) If $\xi_\rho(0) > \xi_\rho(1)$, *then* $\xi_\rho(0) \geq \xi_d(1) > \xi_d(0) \geq \xi_\rho(1)$ *and the equalities hold if and only if* $n = 2k$.

(3) If $\xi_\rho(0) < \xi_\rho(1)$, *then* $\xi_\rho(0) \leq \xi_d(1) < \xi_d(0) \leq \xi_\rho(1)$ *and the equalities hold if and only if* $n = 2k$.

Proof. First consider Part (1). If $\xi_\rho(0) = \xi_\rho(1)$ then Part (1) of Theorem 1 readily gives $\xi_d(0) = \xi_d(1)$, since otherwise it implies $n = k$, which is not allowed. This implies that $\xi_\rho(0) - \xi_\rho(1) = \xi_d(0) - \xi_d(1) = 0$. For the converse, note that if $\xi_\rho(0) - \xi_\rho(1) = \xi_d(0) - \xi_d(1)$ and $\xi_\rho(0) - \xi_\rho(1) \neq 0$, then Part (1) of Theorem 1 gives $n - k = -k \Rightarrow n = 0$, which is absurd. So, the only other possibility is if $\xi_\rho(0) - \xi_\rho(1) = 0$. This readily implies $\xi_\rho(0) = \xi_\rho(1)$.

Next, take Part (2). According to Part (1) of Theorem 1, $\xi_\rho(0) > \xi_\rho(1)$ implies $\xi_d(0) < \xi_d(1)$. Furthermore, Part (2) of the theorem says that if $3n = 2kl$, then $\xi_\rho(0) + \xi_\rho(1) = \xi_d(0) + \xi_d(1) = \frac{4l}{n}$. So we have $\xi_\rho(0) = \frac{\xi_\rho(0)+\xi_\rho(1)}{2} + \frac{\xi_\rho(0)-\xi_\rho(1)}{2} = \frac{2l}{n} + \frac{|\xi_\rho(0)-\xi_\rho(1)|}{2}$, where the last equality holds since $\xi_\rho(0) - \xi_\rho(1)$ is positive. Similarly we can get $\xi_\rho(1) = \frac{2l}{n} - \frac{|\xi_\rho(0)-\xi_\rho(1)|}{2}$. An analogous procedure shows $\xi_d(0) = \frac{2l}{n} - \frac{|\xi_d(0)-\xi_d(1)|}{2}$, which is true since $\xi_d(0) - \xi_d(1)$ is negative, and likewise $\xi_d(1) = \frac{2l}{n} + \frac{|\xi_d(0)-\xi_d(1)|}{2}$. Now, from Part (1) of Theorem 1, $\left|\frac{\xi_\rho(0)-\xi_\rho(1)}{\xi_d(0)-\xi_d(1)}\right| = \frac{n-k}{k}$. This gives us $|\xi_\rho(0) - \xi_\rho(1)| \geq |\xi_d(0) - \xi_d(1)|$, if $k \leq \frac{n}{2}$. Now, $k > \frac{n}{2}$ is not permissible, as this implies $l < 3$ when put in $3n = 2kl$, which violates the condition imposed by the first-half challenge (since it requires $l \geq 3$). Therefore, this gives us $\frac{|\xi_\rho(0)-\xi_\rho(1)|}{2} \geq \frac{|\xi_d(0)-\xi_d(1)|}{2}$ and the equality holds if and only if $n = 2k$. This immediately leads to the result we want to prove.

The proof of Part (3) is analogous to that of Part (2) with the inequalities reversed. □

Corollary 2. *Let* $3n = 2kl$. *If* $\xi_\rho(0) - \xi_\rho(1) = \xi_d(0) - \xi_d(1)$, $\xi_\rho(0) = \xi_d(0)$ *and* $\xi_\rho(1) = \xi_d(1)$ *then* $\xi_\rho(0) = \xi_\rho(1) = \xi_d(0) = \xi_d(1) = \frac{2l}{n}$.

Proof. This follows immediately from Part (1) of Corollary 1 and Part (3) of Theorem 1. □

Lemma 1. *If* $3n = 2kl$ *and* $\xi_\rho(0) = \xi_\rho(1) = \xi_d(0) = \xi_d(1) = \frac{2l}{n}$ *then* $p(0, m_1) = p(2, m_2)$, *where*

$$p(0, m_1) = p_0 + p_4 + p_8 + \cdots + p_{m_1},$$
$$p(2, m_2) = p_2 + p_6 + p_{10} + \cdots + p_{m_2},$$

$m_1 = \lfloor \min(k, l)/4 \rfloor$, $m_2 = 2 + \lfloor (\min(k, l) - 2)/4 \rfloor$,[4] *and the* p_i *are as defined in Eq. (4).*

Proof. The proof is given in the full edition of this paper. □

[4] For instance, if $k = 14$ and $l = 15$, then $m_1 = 12$ and $m_2 = 14$.

We can show with the help of a computer simulation (using infinite precision) that none of the values of n, k and l, where $n \leq 5000$, meeting the condition $3n = 2kl$, satisfy $p(0, m_1) = p(2, m_2)$. Beyond this value of n, $p(0, m_1) \neq p(2, m_2)$ still seems to hold, but due to time constraints we could not verify this. Regardless of this limitation, a value of n larger than 5000 seems impractical. In fact, the usability threshold is arguably much lower than 5000. This is true since k needs to be small due to human memory constraints. Let us say that the value of k is desired to be no more than 30. Then we get $l = 250$ which means a window size of 500. It remains an open problem to obtain an analytical proof of the following lemma for all values of n.

Lemma 2. *If* $3n = 2kl$, *then* $p(0, m_1) \neq p(2, m_2)$ *for all values of* $n \leq 5000$.

Proof. Computer assisted. See Appendix A for the algorithm used to verify this. □

The above lemma tells us that for all practical values of (n, k, l) the following always hold:

1. $\xi_\rho(0) - \xi_d(0) \neq 0$,
2. $\xi_\rho(1) - \xi_d(1) \neq 0$,
3. $(\xi_\rho(0) - \xi_\rho(1)) - (\xi_d(0) - \xi_d(1)) \neq 0$.

Any of the three inequalities above can be used for 1D-RDFA. We shall refer to the left-hand terms in these inequalities as *differentials*. Moreover, since the last differential involves two response values instead of just one, we should call it the *two-response differential* to distinguish it from the *one-response differentials*. Among the three, the two-response differential has the best differentiation capability. To see this, first assume that $\xi_\rho(0) > \xi_\rho(1)$. Then $(\xi_\rho(0) - \xi_\rho(1)) - (\xi_d(0) - \xi_d(1)) = (\xi_\rho(0) - \xi_d(0)) + (\xi_d(1) - \xi_\rho(1))$. Since the expressions $\xi_\rho(0) - \xi_d(0)$ and $\xi_d(1) - \xi_\rho(1)$ are positive, according to Part (2) of Corollary 1, it follows that $(\xi_\rho(0) - \xi_\rho(1)) - (\xi_d(0) - \xi_d(1))$ is greater than the first two differentials. Similarly, if $\xi_\rho(0) < \xi_\rho(1)$, we get $(\xi_d(0) - \xi_d(1)) - (\xi_\rho(0) - \xi_\rho(1)) = (\xi_d(0) - \xi_\rho(0)) + (\xi_\rho(1) - \xi_d(1))$. Once again the two expressions on the right are positive, leading to the desired conclusion. Therefore, the two-response differential will be used to explain how 1D-RDFA works. The attack is based on counting the occurrence of all objects in a fixed number of observed authentication sessions and storing the numbers corresponding to 0-responses and 1-responses separately. For each object we define the difference between its occurrence in 0-response challenges and that in 1-response challenges as its *frequency difference*. Then we rank all the objects according to their frequency differences, and pick the top k ranked objects as the recovered k pass-objects. We have assumed here that $\xi_\rho(0) - \xi_\rho(1) > \xi_d(0) - \xi_d(1)$. If the reverse is true then the attack picks the bottom k ranked objects. Our simulated attacks showed that this 1D-RDFA can recover around half of the pass-objects with around 1,000 observed authentication sessions and all pass-objects with about 7,000 observed authentication sessions using the default parameter values $(n, k, l) = (140, 14, 15)$. While the number of authentication sessions is large, the attack does show the theoretical insecurity of the Foxtail protocol for long-term use without password renewal.

From the imbalance between the expected occurrence of pass-objects and decoy objects in 1D space, one can easily extend the attack to 2D space by considering pairs of objects. This is exactly what Yan et al. demonstrated in their paper [27]. By checking object pairs, the chance of decoy objects getting ranked higher than pass-objects becomes even lower compared with the 1D case, so the attack allows recovering the password with a much smaller number of observed authentication sessions. Figure 1 shows the results of a simulated 2D attack, which agrees well with the results reported in [27].

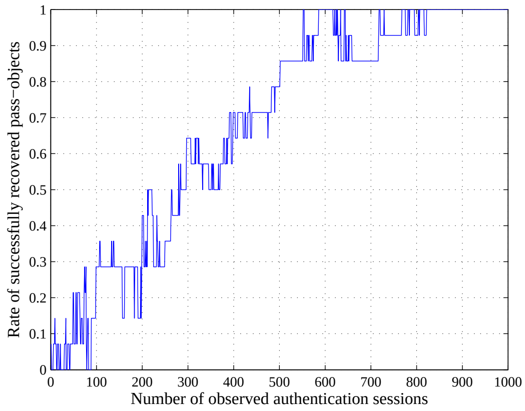


Figure 1. The rate of successfully recovered pass-objects w.r.t. the number of observed authentication sessions.

5. Generalized Frequency Analysis

So far we have discussed 1D and 2D-RDFA. This can be generalized to any $\delta \in \{1, \ldots, k\}$ in a straightforward manner. We have also seen that the condition $3n = 2kl$ is imposed on Foxtail to prevent 1D-RIFA. As it turns out, this condition only precludes 1D-RIFA, and multi-dimensional RIFA is still possible. To understand this, we extend the notion of expected occurrences developed in the previous section to include response-independence. Specifically, we allow $\mathcal{R}$ to take on another value, symbolized by ∞, which indicates that the response is irrelevant. Then

$$\begin{aligned} F_\infty(c_1) &= \underset{\mathcal{R}}{\cup} F_{\mathcal{R}}(c_1) = F_0(c_1) \cup F_1(c_1) \\ &= \{c_2 \,| c_2 \in \{0, \ldots, \min(k,l)\}\} \\ &= \{0, \ldots, \min(k,l)\}. \end{aligned}$$

Since $F_\infty(c_1)$ is independent of c_1, it is natural to denote this simply as F_∞. Now for any $\delta \in \{1, \ldots, k\}$ and for any $\mathcal{R} \in \{0,1\} \cup \infty$, we have

$$\xi_\rho^\delta(\mathcal{R}) = \frac{1}{\tilde{p}_{\mathcal{R}}} \sum\nolimits_{c_1=0}^{3} \sum\nolimits_{c_2 \in F_{\mathcal{R}}} p'_{c_1} p_{c_2} q(c_1, c_2), \qquad (6)$$

and

$$\xi_d^\delta(\mathcal{R}) = \frac{1}{\tilde{p}_{\mathcal{R}}} \sum\nolimits_{c_1=0}^{3} \sum\nolimits_{c_2 \in F_{\mathcal{R}}} p'_{c_1} p_{c_2} q'(c_1, c_2). \qquad (7)$$

If $\delta = 1$, we shall often ignore the superscript to be consistent with the notation used previously. In the above, $q(c_1, c_2)$ is the probability that a pass-object δ-tuple occurs in a challenge containing c_1 and c_2 pass-objects in the first and second-half challenges, respectively. $q'(c_1, c_2)$ is defined similarly for a decoy object δ-tuple.[5] In the previous section, we showed explicit equations of $q(c_1, c_2)$ and $q'(c_1, c_2)$ for the case $\delta = 1$ and $\mathcal{R} \in \{0, 1\}$. The expressions are the same for $\delta = 1$ and $\mathcal{R} = \infty$. Unfortunately, for higher values of δ these probabilities are not easily expressible since a large number of cases has to be considered which increases with δ. However, we can still estimate these quantities empirically using Monte Carlo simulations. Thus, we shall be using experimental estimates when considering $\delta > 1$.

Now if $3n = 2kl$, then $\xi_\rho^1(\infty) = \xi_d^1(\infty)$, which implies that 1D-RIFA will not be fruitful. But this does not necessarily imply $\xi_\rho^\delta(\infty) = \xi_d^\delta(\infty)$ for $\delta \geq 2$. For instance, the values $(n, k, l) = (20, 5, 6)$, although artificially small, satisfy $3n = 2kl$ implying $\xi_\rho^1(\infty) - \xi_d^1(\infty) = 0$. Yet, by simulating 100,000 Foxtail challenge-response pairs, we find that $\xi_\rho^2(\infty) - \xi_d^2(\infty) = 0.01256$ which is significantly larger than 0. Thus, a 2D-RIFA is plausible. In 100 trials, we were able to obtain the password in 386 sessions on average with these parameters. By contrast 2D-RDFA with two-response differential, i.e., using the difference $(\xi_\rho^2(0) - \xi_\rho^2(1)) - (\xi_d^2(0) - \xi_d^2(1))$, was able to find the secret in merely about 13 sessions. Although the performance of 2D-RIFA is poorer, the results show that the attack is possible nonetheless.

It is worthwhile to see if higher dimensional frequency analysis reveals the password in fewer sessions. To this end, we ran a Monte Carlo simulation with the values $(n, k, l) = (140, 14, 15)$, to estimate $\xi_\rho^\delta(\mathcal{R})$ and $\xi_d^\delta(\mathcal{R})$ by randomly sampling 100,000 challenge-response pairs. Based on these results, the three differentials mentioned in the previous section together with the *no-response differential* $\xi_\rho^\delta(\infty) - \xi_d^\delta(\infty)$ are plotted in Figure 2, as δ ranges from 1 to k. As is evident, the differentials are non-negligible uptil $\delta = 4$, after which they approach 0. The figure suggests that higher dimensional frequency analysis (beyond $\delta = 4$), in general, might not be practical in finding the password since the difference $\xi_\rho^\delta(\mathcal{R}) - \xi_\rho^\delta(\mathcal{R})$ approaches 0 rapidly with increasing δ. This also makes intuitive sense; as δ increases the expected occurrence of a δ-tuple also decreases (more objects need to be in a challenge at the same time), this in turn increases the number of samples required to distinguish between pass and decoy object δ-tuples. The smaller the value of the differential, the larger the number of challenge-response pairs required to reveal the password. Together with Lemma 2, this also shows that it is not possible to ensure $\xi_\rho^\delta(\mathcal{R}) = \xi_d^\delta(\mathcal{R})$ holds for all δ and all $\mathcal{R}$ in the Foxtail protocol.

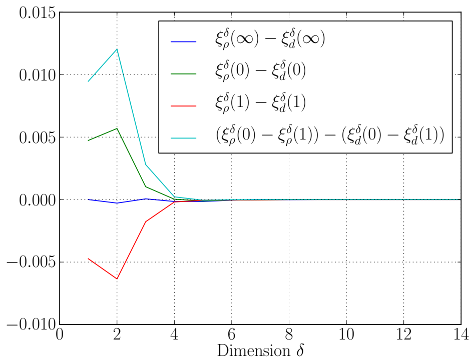


Figure 2. The no-response, one-response and two-response differentials as d ranges from 1 to k.

In practice, however, negligible values of the differentials are sufficient, as it will take a huge number of challenge-response pairs to obtain the password. It is important, then, to find a theoretical estimate of the number of pairs required to distinguish between a pass and decoy object δ-tuple with a given probability. Let us define an indicator random variable $I_\rho^\delta(i, \mathcal{R})$ which is 1 if a pass-object δ-tuple occurs in a challenge. $I_d^\delta(i, \mathcal{R})$ is defined likewise. To illustrate the estimation method, we consider 2D-RDFA with two-response differential. The case for other variants is similar. Two-response 2D-RDFA examines the following probability:

$$\Pr\left[\sum\nolimits_{i=0}^{m} (I_\rho^\delta(i, 0) - I_\rho^\delta(i, 1))\right.$$

[5] We do not consider tuples which are composed of both pass and decoy objects for two reasons. First it makes the analysis increasingly complex. Second, if $\xi_\rho^\delta(\mathcal{R}) = \xi_d^\delta(\mathcal{R})$ holds for all δ, then the expected occurrences of the *mixed* δ-tuples also tend to be the same.

$$> \sum\nolimits_{i=0}^{m} (I_d^\delta(i,0) - I_d^\delta(i,1))\Big].$$

We have assumed that the frequency difference of the pass-object δ-tuples is higher than that of decoy object δ-tuples. If the opposite is true, the inequality can be duly reversed. Also, it is implicitly assumed that there is an equal number of challenge-response pairs corresponding to 0 and 1-responses at our disposal. This is reasonable since the response is uniformly distributed. Thus, m is in fact half of the total number of pairs observed. Let

$$I_1 = I_\rho^\delta(i,0),\ I_2 = I_\rho^\delta(i,1),\ I_3 = I_d^\delta(i,0) \text{ and } I_4 = I_d^\delta(i,1).$$

Define

$$Z_{\rho,d}^\delta(i,\mathcal{R}) = I_1 - I_2 - I_3 + I_4 = a_1 I_1 + a_2 I_2 + a_3 I_4 + a_4 I_4,$$

where $a_1 = 1, a_2 = -1, a_3 = -1$ and $a_4 = 1$. The expected value of $Z_{\rho,d}^\delta(i,\mathcal{R})$ is

$$\begin{aligned} \mathrm{E}[Z_{\rho,d}^\delta(i,\mathcal{R})] &= a_1 \mathrm{E}[I_1] + a_2 \mathrm{E}[I_2] + a_3 \mathrm{E}[I_3] + a_4 \mathrm{E}[I_4] \\ &= \xi_\rho^\delta(0) - \xi_\rho^\delta(1) - \xi_d^\delta(0) + \xi_d^\delta(1). \end{aligned}$$

Its variance is

$$\mathrm{Var}[Z_{\rho,d}^\delta(i,\mathcal{R})] = \sum_{j=1}^{4} a_j^2 \mathrm{Var}[I_j] + 2 \sum_{j=1}^{4} \sum_{k>j} a_j a_k \mathrm{Cov}[I_j, I_k],$$

where $\mathrm{Cov}(\cdot,\cdot)$ is the covariance. Denote $\mathrm{E}[Z_{\rho,d}^\delta(i,\mathcal{R})]$ by μ_Z and $\mathrm{Var}[Z_{\rho,d}^\delta(i,\mathcal{R})]$ by σ_Z^2. The aforesaid probability translates into

$$\begin{aligned} &\Pr\left[\sum\nolimits_{i=0}^{m} Z_{\rho,d}^\delta(i,\mathcal{R}) > 0\right] \\ &= \Pr\left[\sum\nolimits_{i=0}^{m} Z_{\rho,d}^\delta(i,\mathcal{R}) - m\mu_Z > -m\mu_Z\right] \\ &= \Pr\left[\frac{\sum_{i=0}^{m} Z_{\rho,d}^\delta(i,\mathcal{R}) - m\mu_Z}{\sigma_Z \sqrt{m}} > -\frac{\sqrt{m}\mu_Z}{\sigma_z}\right] \\ &\approx \Phi\left(-\frac{\sqrt{m}\mu_Z}{\sigma_z}\right), \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal random variable. The approximation follows from the central limit theorem [18, §8.3, p. 434]. Now, say we let $\Pr\left[\sum_{i=0}^{m} Z_{\rho,d}^\delta(i,\mathcal{R}) > 0\right] = 0.6$, then we can estimate the number of pairs required as

$$\begin{aligned} \Phi\left(-\frac{\sqrt{m}\mu_Z}{\sigma_z}\right) &= 0.6 \\ \Rightarrow -\frac{\sqrt{m}\mu_Z}{\sigma_z} &= 0.25335 \\ \Rightarrow m &= \frac{0.06419\sigma_Z^2}{\mu_Z^2}. \end{aligned}$$

Table 1. Number of pairs required such that $\Pr[Z_{\rho,d}^\delta(i,\mathcal{R}) > 0] = 0.6$, with $n = 140$, $k = 14$ and $l = 15$.

δ	1	2	3	4	5
m_{lb} = pairs	1095	136	489	14144	47742
s_{lb} = sessions	55	7	24	707	2387
μ_Z	0.00947	0.01204	0.00280	0.00022	-0.00005
σ_Z^2	0.76527	0.15345	0.02987	0.00533	0.00093

As discussed before, since m is half of the challenge-response pairs observed, the total number of pairs required is double this amount. Let us denote this by m_{lb}, i.e., $m_{\mathsf{lb}} = 2m$, where lb stands for lower bound. Similarly, we denote the number of sessions corresponding to m_{lb} by s_{lb}. With the values $(n, k, l) = (140, 14, 15)$ we calculated the number of samples required so that the probability of distinguishing a pass-object δ-tuple from a decoy-object δ-tuple is at least 0.6. The results are shown in Table 1. Beyond $\delta = 5$ the numbers are negligible, and as such are not shown in the table. We obtained these results with the help of a Monte Carlo simulation, estimating the expected value and variance of $Z_{\rho,d}^\delta(i,\mathcal{R})$. While this leaves room for slight errors, we can still notice a trend, namely as δ grows beyond its first few values, a very high number of samples are required to distinguish between the two types of tuples.

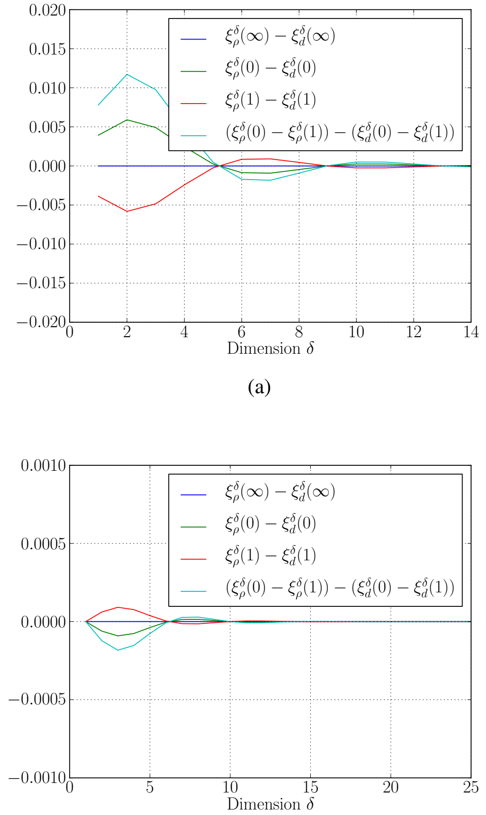
Note that these results do not mean that the adversary can obtain the entire password after these many samples, or even obtain a single pass-object δ-tuple. This only serves as a lower bound, which is a loose but safe. For instance, it takes on average about 711 sessions to find the whole password using 2D-RIFA with these parameter values [27] whereas the corresponding value of s_{lb} in Table 1 is only 7. But in practice, it might not be possible to find the average number of sessions required to find the password using δ dimensional frequency analysis for each $\delta \leq k$ through simulations, since the computational burden increases considerably. Under such circumstances, one can use m_{lb} or s_{lb} as a safe lower bound to limit the number of sessions a protocol can be used with the same password. As a marker, at least 2,000 authentication sessions is reasonable. An estimate of the number of challenge-response pairs required to obtain the whole password remains an open problem. It appears that no practical values of the parameters in Foxtail can ensure $s_{\mathsf{lb}} > 2000$ for all δ. As an example, with the parameters $(n, k, l) = (160, 24, 10)$, for which a brute-force attack has time complexity greater than 2^{80} [13], two-response 2D-RDFA was able to find the password in a little more than 1,200 sessions on average in 10 trials.

6. Is the Challenge Generation Method the Problem?

It is interesting to ponder whether the method of generating challenges in counting based protocols is the reason for the disparity in the differentials. To analyze this we consider three variants of Foxtail. The first variant of Foxtail does not contain a window, and we duly call this the *no-window* Foxtail. A challenge in this variant is generated by sampling each object uniformly at random, i.e., each object is included in the challenge with probability $1/2$. Another way of looking at this is that now we have a variable length window. The other two variants are derived from the way the two half-challenges are generated in the original Foxtail. We name these variants the Uni-rule and the Rand-rule Foxtail after the way the challenges are generated. Both these variants have a fixed window size of l. Note that there are no half-challenges used for all these variants of Foxtail. Notice that the combination of the two challenges in the original Foxtail was there to circumvent a partially known password attack, which is a much stronger notion of security not considered in this paper.

For comparison, the formulas for the differentials are shown in Table 2. As an example, the derivation of the results for no-window Foxtail is shown in Appendix B. The derivation for the other two is more involving and is given in the full version of the paper. The parameter λ in Uni-rule Foxtail is a fixed positive integer $\leq k+1$. In the original Foxtail, it was fixed at 4 for the first half-challenge. The parameter ensures that the probability of a 0-response is the same as the probability of a 1-response. As a result the quantities $\tilde{p}_0$ and $\tilde{p}_1$ are $\frac{1}{2}$ in case of Uni-rule Foxtail. This is also true in case of the original Foxtail (The proof is in the full edition of the paper). This is not the case with the no-window and Rand-rule Foxtail. The resulting bias means an increased success probability of random guess. However, for reasonably large values of protocol parameters the difference in probabilities is too small for random guess to have any advantage. On the other hand, these two variants of Foxtail do not require any condition to ensure that the no-response differential is 0 for $\delta = 1$. In fact, in contrast with the Uni-rule and original Foxtail, the no-response differential is 0 for all δ. See Appendix B for the proof of this result for the no-window Foxtail. The proof for the Rand-rule Foxtail is much longer and is given in the full edition of the paper.

On the other hand, the response based differentials are not necessarily zero. An interesting observation is that the differentials in no-window Foxtail are only dependent on k, and not on n. Thus, it is possible to minimize the differentials by increasing k alone. Due to the absence of half-challenges, the expression for higher-dimensional differentials is straightforward in all these variants of Foxtail. Figure 3 shows these differentials for the no-window Foxtail. As the figure suggests, when $n = 140$ and $k = 14$ the lower-dimensional differentials are non-negligible, but with $k = 25$ they are very small. Note that the y-axis scale is different in the two plots. Table 3 shows the number of sessions required to distinguish a pass-object δ-tuple from its decoy object counterpart (i.e., s_{lb}). As can be seen, s_{lb} is quite large. This is not the case with $k = 14$. For instance, with $\delta = 2$, the two-response differential requires only 35 sessions to distinguish. Notice that the original Foxtail is still not secure against frequency analysis with the comparative value of $k = 24$ as mentioned before, with $n = 160$ and $l = 10$.



(a)

(b)

Figure 3. The differentials for no-window Foxtail with (a) $n = 140$ and $k = 14$, (b) $n = 140$ and $k = 25$.

We can conclude from this that the no-window Foxtail is safe from frequency analysis with $n = 140$ and $k = 25$, since it can be used for a sufficiently large number of sessions. However, this is still not satisfactory for two main reasons. First, although it can be used for a huge number of sessions (at least 64,000), we would like to see if there is a

Table 2. Comparison of differentials in the three variants of Foxtail.

	No-window	Rand-rule	Uni-rule
$F_{\mathcal{R}}$	$\{c \lvert \lfloor \frac{c \bmod 4}{2} \rfloor = \mathcal{R}\}$	$\{c \lvert \lfloor \frac{c \bmod 4}{2} \rfloor = \mathcal{R}\}$	$\{c \lvert \lfloor \frac{c \bmod 4}{2} \rfloor = \mathcal{R}\}$
$\tilde{p}_{\mathcal{R}}$	$\sum_{c \in F_{\mathcal{R}}} \binom{k}{c} \frac{1}{2^k}$	$\sum_{c \in F_{\mathcal{R}}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}}$	$\sum_{c \in F_{\mathcal{R}}} \frac{1}{\lambda}$
$\xi_\rho^\delta(\mathcal{R}) - \xi_d^\delta(\mathcal{R})$	$\frac{1}{\tilde{p}_{\mathcal{R}}} \sum_{c \in F_{\mathcal{R}}} \binom{k}{c} \frac{1}{2^k} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{1}{2^\delta} \right)$	$\frac{1}{\tilde{p}_{\mathcal{R}}} \sum_{c \in F_{\mathcal{R}}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}} \right)$	$\sum_{c \in F_{\mathcal{R}}} \frac{1}{\lambda} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}} \right)$
$\xi_\rho^\delta(\infty) - \xi_d^\delta(\infty)$	$\sum_{c=0}^{k} \binom{k}{c} \frac{1}{2^k} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{1}{2^\delta} \right) = 0$	$\sum_{c=0}^{\min\{k,l\}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}} \right) = 0$	$\sum_{c=0}^{\min\{k,l\}} \frac{1}{\lambda} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}} \right)$ 0 only if $\lambda = k+1$, $n = 2l$ and $k = l$

Table 3. Number of pairs in no-window Foxtail to ensure $\Pr[Z_{\rho,d}^\delta(i, \mathcal{R}) > 0] = 0.6$, with $n = 140$ and $k = 25$.

δ	1	2	3	4	5
m_{lb} = pairs	∞	6,461,048	1,675,087	1,292,210	2,670,567
s_{lb} = sessions	∞	323,052	83,754	64,610	133,528
μ_Z	0	0.000122	0.000183	0.000152	0.000076
σ_Z^2	1.0	0.750000	0.437500	0.234375	0.121093

way to completely remove the threat of frequency analysis no matter how many sessions are observed. Secondly, the values of parameters for which the protocol is safe are arguably not practical. The introduction of a window in the original Foxtail ensures that the number of objects in a challenge are fixed, i.e., $2l$. This reduces the cognitive load on the user searching for his pass-objects if l is small. The absence of a window implies that a much larger number of objects can be present in each challenge and if $k = 25$, a much larger number of pass-objects too, thus making it harder for the user to locate his pass-objects. This can be solved by making k smaller, say 10 to 14. But, as we have seen, this value of k is not secure against frequency analysis.

As noted before, the no-window and Rand-rule Foxtail are inherently resistant to RIFA, since the no-response differentials are zero for all dimensions. Intuitively, this is due to the fact that the challenges in both these variants do not differentiate between pass and decoy objects. Both types are drawn at random with the same probability. This is not true in the case of Uni-rule Foxtail. There are two cases depending on the value of λ. First, if $\lambda < k + 1$, then a maximum of $\lambda - 1$ pass-objects are present in a challenge. Thus, a λD-RIFA can be carried out since there is no pass-object λ-tuple in the challenge. The second case is when $\lambda = k + 1$. This results in the conditions $n = 2l$ and $l = k$ which are necessary to ensure that the no-response differentials are zero for all dimensions. Unfortunately, these restrictions deem the protocol impractical, since n needs to be large enough to prevent brute-force attack which means k, being half of n, is going to be large too. This implies that in theory, it is not possible to completely eradicate the threat of multi-dimensional RIFA in the Uni-rule and original Foxtail, although in practice this might require a large number of sessions. One way to make the differentials negligible is to increase the values of n and k (see the equations of the differentials), but this increases cognitive load.

7. Frequency Analysis on Non-counting based Protocols

Yan et al. also showed that frequency analysis can be applied on non-counting based protocols that use some structure in the challenge [27, §4.2]. Consider for example, the CHC protocol from [25]. In this protocol, the user responds by clicking inside the convex hull of any three of the pass-objects present in the challenge. The protocol ensures that at least 3 objects are present in each challenge, so that a convex hull can be formed. In [25], the window-size was variable with an average value of $l = 83$. Yan et al. showed that 2D-RIFA reveals the password after the observation of 18 challenge response pairs or less than 2 sessions, with the parameter values $(n, k, l) = (112, 5, 83)$. However, it was shown in [1] that the parameters should satisfy the relation $n = \frac{2kl}{k+3}$ to prevent 1D-RIFA. Under this relation, the aforementioned set of parameter values is not recommended. A comparable set of recommended values is $(n, k, l) = (115, 5, 92)$. But as we shall see, these set of values still do not completely eradicate the threat of higher-dimensional RIFA. In a manner analogous to the derivation of expected occurrences for Foxtail, we see that for CHC

$$\xi_\rho^\delta(\infty) = \sum_{c=3}^{k} \frac{1}{k-2} \frac{\binom{c}{\delta}}{\binom{k}{\delta}}, \text{ and } \xi_d^\delta(\infty) = \sum_{c=3}^{k} \frac{1}{k-2} \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}},$$

where we have assumed that the challenge is generated such that $3, 4, \ldots, k$ pass-objects are present with the same probability [1, 25]; hence the term $\frac{1}{k-2}$. With the above set of values the 2D no-response differential, i.e., $\xi_\rho^2(\infty) - \xi_d^2(\infty)$, is -0.00525. This corresponds to an m_{lb} of 2153.[6] On the other hand, with $(n, k, l) = (112, 5, 83)$, $\xi_\rho^2(\infty) - \xi_d^2(\infty)$ is 0.08998 giving an m_{lb} of just over 7. Thus, if the aforementioned relation is satisfied by the parameter values, CHC is relatively safe from 2D-RIFA in the sense that the attacker has to observe a larger number of session. However, in theory it is still susceptible to the attack.

Another protocol with an apparent structure in its challenge is Undercover [19], which requires that at most one pass-object be present in a challenge. If the challenges are generated randomly, then this piece of information can be used to find the password in less than 10 sessions [17, 27]. This can be explained with the help of the differentials. We see that $\xi_\rho^\delta(\infty) = 0$ for all $\delta > 1$. On the other hand $\xi_d^\delta(\infty)$ is non-zero for any $\delta > 1$. This implies that the corresponding differentials are non-zero. Thus, δD-RIFA for any $\delta > 1$ can be carried out to find the password. We have seen that the same problem can also arise in the Uni-rule Foxtail discussed in the previous section, if a value of $\lambda < k + 1$ is used. In such a case the protocol is susceptible to λD-RIFA, which can be practical if λ is too low.

Yan et al. [27] also remarked that the APW protocol [2] does not succumb to frequency analysis. We can see why APW is resistant to frequency analysis. APW is a non-counting based protocol in which each object has a random integer *weight* associated with it from 0 to 9. This is equivalent to saying that each object is drawn at random with probability $9/10$, i.e., an object occurs in a challenge if its weight is non-zero. The APW does not use a window, which means that it is safe from RIFA since the differentials are all zero which can be verified in a manner similar to the case of no-window Foxtail. The protocol is also safe from RIFA if it uses a window and the challenges are generated using the Rand-rule method. However, if the protocol is to use the Uni-rule method of generating the challenge, RIFA is applicable.

Since the CHC and APW protocols are not based on counting, RDFA does not straightforwardly apply to them. On the other hand, from these examples, we see that RIFA is applicable to non-counting based protocols as well, as long as the challenge generation method distinguishes between pass and decoy objects. The severity of this vulnerability can be empirically determined by finding the value of s_{lb}. A very low value of s_{lb} means that the corresponding set of parameter values are insecure. An important remark is that resistance to frequency analysis does not imply that the protocol is secure from all other attacks. For instance, with $(n, k, l) = (115, 5, 92)$, the CHC protocol is safe from RIFA in terms of the number of observed sessions required, yet a brute force attack can easily find the password, since the time-complexity is low ($\approx 2^{27}$).

8. Can Counting based Protocols be Completely Secure against Frequency Analysis?

The preceding discussion reveals that there are two principles necessary to completely eradicate the threat of frequency analysis on counting based protocols.

1. Each object in a challenge should be sampled independently with the same probability regardless of its type. This is to prevent RIFA.

2. The response should be independent of the number of pass-objects present in a challenge. This prevents RDFA (This seems contradictory when applied to counting based protocols, but see the discussion that follows).

The first principle does not mean that parameters be chosen such that the probabilities are *artificially* made the same. This is precisely what was done in the Foxtail protocol resulting in the condition $3n = 2kl$, which prevents 1D-RIFA but not its higher dimensional variants. If, instead, each object is sampled uniformly at random, as in the case of the no-window or Rand-rule Foxtail, RIFA is unsuccessful for all dimensions. This principle is not restricted to counting based protocols. We have already seen how CHC, a non-counting based protocol, is susceptible to RIFA. Yan et al. proposed the same principle for protocols where there is a *structural requirement* for the challenges. Such as, there should be at least 3 pass-icons present in a challenge as in the case of CHC. As it turns out, the principle is important to avoid frequency analysis as well.

The second principle above seems to be related to counting based protocols only. Notice that the responses in a counting based protocol are dependent on the number of pass-objects present in a challenge. Compare this to the APW or the CHC protocol. There, the responses are independent of the number of pass-objects present. In APW, the response is a random integer corresponding to an index of an array of length n. In CHC, the response is a click, whose location is not altered by the number of pass objects present. Therefore both these protocols, and any protocol whose reponse is not dependent on the number of pass objects present in a challenge is safe from RDFA. Thus, a fix needs to be found for counting based protocols to make the response independent of the number of pass-objects present in the challenge. A desirable property of such a fix is that it should be able to retain all the features of a counting based protocol. Notice that these results can be generalized to any

[6] Variance is not shown.

$|\mathcal{R}| > 2$. In such a case, one can still use pair-wise differentials, or more generally, tuple-wise differentials.

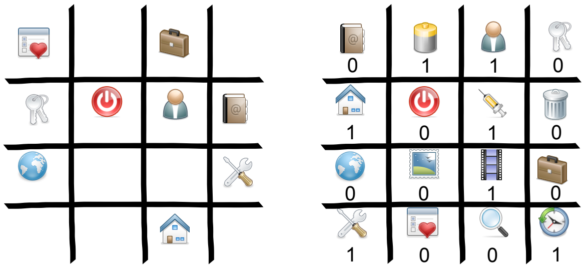
Based on these two principles we first show a fix that can be used for any counting based protocol. We illustrate the fix by once again using Foxtail as an example. The drawback of the fix is that it increases the authentication time almost twofolds. We further show a fix specific to the Foxtail protocol that incurs a smaller time penalty. We also discuss how this fix can be generalized to all counting based protocols.

8.1. A Fix for Counting based Protocols

One of the reasons why RDFA works is that the true challenges and responses are all observable to the attacker. If we can design the protocol to deliver a partially hidden challenge to the user, we may be able to make the true challenge and/or the true response only partly observable to the attacker. For instance, the hidden challenge may be used to skip one or more pass-objects in the challenge as if a different challenge was shown to the user, or it could be used to reverse the response so that the attacker is confused about the true response. Taking the Foxtail protocol as an example, if the hidden challenge can uniformly confuse the two possible responses, RDFA will not work any more because the expected occurrences will be the average of two kinds of challenges, thus reducing it to the case where the response is not considered at all. How to select a proper way of encoding the hidden challenge and avoiding any information leakage about the hidden challenge will be the core of this idea. The idea of hidden challenges was first proposed in [19] using an additional device accessible to the legitimate user only.[7] In this section, we describe a new framework of using *soft hidden challenges* with counting based protocols, where soft hidden challenges are hidden challenges embedded in the challenge without the need of getting any auxiliary hardware involved. The framework can be considered a fix for all counting based protocols against RDFA. We illustrate this in the following using Foxtail as an example.

The first change we introduce in the enhanced Foxtail protocol is that we generate the challenge without considering the type of object, i.e., pass or decoy object. This can be achieved either through a window-based design or a non-window based design. In the first case, the challenge is generated using the Rand-rule, i.e., for the fixed window-size l, a total of l objects are drawn at random from the pool of n objects, regardless of their type, for each challenge. In the case of the no-window Foxtail, each object is present in a challenge with probability $1/2$. As we have seen, this completely eradicates the threat of RIFA. To illustrate the second change, we use the Rand-rule challenge generation method. Notice that there is no condition imposed on the parameters n, l and k for this method. This means that we are free to choose any size of the window. In order to make the protocol safe from the RDFA, the response to the Foxtail function is flipped according to a random bit. We call this random bit, the flip-bit. The flip-bit is not communicated in the open. We will discuss how this can be achieved in a moment. To see why this fix completely prevents RDFA, see Appendix C. The main idea is that the flip-bit ensures that the occurrence of the pass-objects is independent of the response. In a similar way, we can show that the response-based differentials are 0 if the no-window Foxtail is used in conjunction with the flip-bit.

We now return to the implementation of the flip-bit. One way to achieve this is to have two challenges per round. In the first challenge, the user counts the number of pass-objects present and computes the Foxtail function as before. Once the function is computed, a new challenge is shown which has all n objects present. Each object in the challenge has a random bit associated with it. The user locates a *predetermined* pass-object, retrieves the bit associated with it, and sends the response as the $\bmod 2$ sum of the bit and the result of the Foxtail function. Notice that the flip-bit (second) challenge is an instance of soft hidden challenges discussed before. Here, part of the challenge is hidden from the observer in the sense that without knowing the response to the Foxtail function, it is not possible to know which is the flip-bit apart from guessing the pass-object. Figure 4 shows an example of the two types of challenges in the modified Rand-rule Foxtail.



(a) (b)

Figure 4. Modified Foxtail (a) With the first challenge (b) With the flip-bit (second) challenge.

Unfortunately, this simple fix has two drawbacks. First, the attacker can randomly guess the pass-object corresponding to the flip-bit with probability $\frac{1}{n}$, which is quite small since a typical value of n is 140. If the attacker's guess is correct, RDFA can be applied by removing the flip-bit from

[7] Note that the particular design in [19] has some security flaws related to its human-computer interface as pointed out in [17].

the response. The probability of success is $\frac{1}{n}$, which is approximately 0.0071 when n is 140. The success probability of this random guess attack can be reduced by increasing n, however, too large a value of n will incur usability costs. Moreover, there is an even smarter attack which can still work with a large value of n. The attacker can simply treat each of the n bits in the second challenge as the flip-bit, remove it from the response and apply RDFA. All bits other than the flip-bit will show no difference between pass and decoy object occurrences, since they are independent of the response. Thus, the frequency differences will be zero for these bits. However, removal of the flip-bit alone will make the frequency difference reappear, and hence RDFA can be straightforwardly applied.

A possible solution to the above-mentioned security vulnerabilities is to obtain the flip-bit as the $\bmod 2$ sum of bits corresponding to *multiple* pass-objects, which results in a protocol that is computationally secure against RDFA. More specifically, let $k' \leq k$ be the number of pass-objects selected in the setup phase to compute the flip-bit. Then the aforementioned attack has complexity $\mathcal{O}(\binom{n}{k'})$, since the attacker has to go through all k'-combinations of objects in the second challenge. Thus increasing k' increases the time-complexity of the aforementioned version of RDFA, and with $k' = k$ the complexity is the same as brute-force. The drawback is the extra authentication time imposed by the calculation of the flip-bit, which increases with k'. This fix can be generalized to any function and any range of response values. Generally speaking, the flip-bit can be replaced by a random integer modulo $|\mathcal{R}|$ which is computed as the $\bmod |\mathcal{R}|$ sum of $k' \leq k$ pass-objects each carrying a random *weight* from the set $\{0, 1, \ldots, |\mathcal{R}| - 1\}$ in the second challenge. An analysis similar to the one above shows that the differentials are all 0. In particular, this fix can be applied to the function $\#\mathcal{C}(P) \bmod q$ for any $q \geq 2$, and the resulting scheme will be computationally secure against frequency analysis. Yan et al. identified this as an insecure function against frequency analysis [27, §4.4], but by introducing the flip-bit fix the protocol based on this function can be made computationally secure against frequency analysis. Note that this does not mean that a protocol based on the function $\#\mathcal{C}(P) \bmod q$ is free from any other attacks. Gaussian elimination is one example.

There does not appear to be any fix for the Uni-rule and original Foxtail. The evidence in support of this has been given in the previous sections where we considered the original Foxtail and its variants. It also seems that there is no fix for these variants of Foxtail with practical values of parameters that can prevent frequency analysis for sufficiently large s_{lb} (say 2000). However, it is possible to resist frequency analysis for large values of s_{lb} by using a different counting function. Yan et al. point out that the function $\#\mathcal{C}(P) \bmod 2$ is safe from frequency analysis [27, §4.4]. Our results show that the differentials are still non-zero with the parameters $(n, k, l) = (140, 14, 15)$ for this function. The difference appears to be that the differentials are much smaller in this case. The quantity s_{lb} is about 9000 for $\delta = 1$. Compare this with the case of the Foxtail function, which has $s_{\mathsf{lb}} = 55$ for the same δ. Perhaps this led Yan et al. to conclude that using this function will prevent frequency analysis. However, this function itself is not sufficient, since it is vulnerable to an attack based on Gaussian elimination after the observation of a small number of sessions [13, 27].

8.2. A Fix for Foxtail

The flip-bit solution increases the authentication time considerably, since the user has to respond to double the number of challenges per session as compared to the original Foxtail with a reasonably large value of k'. Here, we present a fix specifically for the Foxtail function, and discuss how this can be applied to a wider range of counting functions. It is not possible to generalize this fix without knowing the actual counting function. In the new version of Foxtail, the challenge consists of all n objects. Each object is assigned a random *weight* from the set $\{0, 1, 2, 3\}$. The computation of the Foxtail function is done as before, with the binary weights replaced by $\bmod 4$ weights. We accordingly call this variant of the protocol, the no-window Foxtail with 4-ary weights. First notice that RIFA will not work on this protocol due to the same reasons detailed in the binary weights case. The protocol is also secure from any RDFA with dimension less than or equal to $k - 1$. To see this, suppose $k = 2$, and consider 1D-RDFA. The response is independent of the weight of a single pass-object since it depends on the weight of the other pass-object which in turn has a random weight from $\{0, 1, 2, 3\}$. Thus, the response does not reveal any bias in the differentials. Notice that, this bias will be there if we consider $k = 1$. In this case, we do not have another pass-object to randomize the response. For higher dimensional RDFA, we can consider a pass-object tuple to be present if *all* its constituent objects have a non-zero weight. Here again, if we consider $k = 3$ and 2D-RDFA, the response randomizes over the individual occurrence of the pass-object pairs. This is true since the response is dependent on the weight of a third pass-object which is randomly chosen from $\{0, 1, 2, 3\}$. Again, there will be bias in the differentials if we consider $k = 2$. In general, for a fixed k, the only version of RDFA that will work is kD-RDFA. But the complexity of such an attack is the same as brute force attack.

We could also propose Rand-rule Foxtail with 4-ary weights. However, this version is not secure against RDFA in the above mentioned sense. To see this, consider Rand-rule Foxtail with 4-ary weights with the parameters

$(n, k, l) = (140, 14, 20)$. Since l is small, with high probability the challenge does not contain more than 1 pass-object. Thus, 1D-RDFA can be applied, since in most challenges a second pass-object is not present to randomize the response. This probability, however, can be made increasingly small by incrementing l. A suitable set of parameters is $(n, k, l) = (140, 14, 40)$, for which our previous criterion of $s_{\mathsf{lb}} > 2000$ applies for all $\delta \geq 2$. However, for $\delta = 1$, the value of s_{lb} is lower than this mark. Yet, through Monte Carlo simulations, we were only able to find the password in more than $16,000$ sessions on average in 100 runs. Thus, the these parameter values are safe against RDFA. It should be noted that this means that the protocol is not completely secure from RDFA, but is safe for a sufficiently large number of sessions. In general, this fix is not straightforwardly applicable to all counting based protocols, as it depends on the specific function. But a general rule can be established that if a counting functions uses a $\bmod q$ operation on the result of counting for $q > 2$, then the objects should be assigned random weights from the set $\{0, 1, \ldots, q-1\}$. Otherwise, the fact that the response is not uniformly distributed can be used in RDFA to find the password.

8.3. Usability Analysis

Based on the quantitative framework introduced by Yan et al. [27, §6], we can also estimate the usability of the *fixed* Foxtail protocols. But since the Rand-rule Foxtail with 4-ary weights is the most usable one, we mainly focus on this variant for the usability analysis. In each round, the protocol requires recognizing k pass-objects, adding their weights, a modulo operation and a small division. Let RT denote the reaction time. Then

$$\mathsf{RT} = (0.3694 + 0.0383k)\left\lceil \frac{l}{4} \right\rceil + \left(\left\lceil \frac{kl}{n} \right\rceil - 1\right) t_0 + t_0 + t_1.$$

The equation $0.3694 + 0.0383k$ is the reaction time for recognition when k pass-objects constitute the *positive set* [22, 27]. Since the window size is l, we have $\frac{l}{4}$ parallel recognition channels, since each individual has up to a limit of 4 parallel recognition channels [27]. t_0 and t_1 represent the reaction time of large addition[8] and small division respectively. The values used here are $t_0 = 0.924$ and $t_1 = 0.959$ [5, 27]. Since $\frac{kl}{n}$ pass-objects are present in a challenge on average, the average number of additions is $\lceil \frac{kl}{n} \rceil - 1$. Hence the average reaction time for adding weights is $(\lceil \frac{kl}{n} \rceil - 1)t_0$.

Plugging in the values $n = 140$, $k = 14$, $l = 40$, $t_0 = 0.924$ and $t_1 = 0.959$ in the equation for RT, we see that the reaction time amounts to about 13.71 seconds per round or 274.22 seconds per session. In comparison, the reaction time is 212.76 seconds per session for the original Foxtail according to Table 1 in [27]. The time is about 3 seconds per round more than the time required in the original Foxtail. This is due to the fact that the simpler task of counting is replaced with large additions. In the case of the enhanced no-window Foxtail with 4-ary weights, the reaction time is more than double. This is due to the fact that the average number of objects present in a challenge is 70 (when n is 140), which is more than double the window size in the original Foxtail. We acknowledge the lack of an actual user study to obtain RT, but we did not find it necessary since we do not claim that the enhanced Foxtail protocols are practical (as is the case with the protocols discussed by Yan et al. in [27]).

9. Conclusion

In this paper, we gave a detailed and quantitative explanation of a specific attack on user authentication protocols called frequency analysis. The attack was introduced by Yan et al. in [27] where they named it counting based statistical analysis. They demonstrated the weakness of several protocols including Foxtail from [13] using this attack. A reduced form of the attack was already known in literature, but its full potential was unknown before the treatment from Yan et al. We extended their work by giving a more quantitative analysis of the attack. We showed that one variant of the attack is applicable to all protocols that generate challenges with the knowledge of the pass and decoy objects, and the attack works even if the parameter values for a given protocol are chosen such that the number of pass and decoy objects occur with the same probability. We described a method to determine the number of sessions a protocol can be safely used with the same password before frequency analysis comes into play. We have also shown that it is not necessary that a candidate function completely eradicate the threat of frequency analysis. In practice, if frequency analysis requires a very high number of sessions, it counts as a sufficient defence. This number can be determined empirically using the method shown in this paper. We have shown a fix that renders any counting based protocol practically safe from frequency analysis. The resulting fix, however, incurs almost double the cost in reaction time. We also described another fix specific to the Foxtail protocol, which causes a lesser increase in reaction time, yet makes the protocol practically secure from frequency analysis. From the usability point of view, it will be helpful to extend the framework from Yan et al. to include reaction times for tasks specific to user authentication protocols. At present, the values for the reaction times are taken from studies that are carried out by experimental psychologists under conditions different from user authentication protocols.

[8] Large addition is defined as the operation in which the product of the two operands is greater than 25 [5].

References

[1] H. J. Asghar, S. Li, J. Pieprzyk, and H. Wang. Cryptanalysis of the Convex Hull Click Human Identification Protocol. In *ISC '10*, pages 24–30. Springer-Verlag, 2010.

[2] H. J. Asghar, J. Pieprzyk, and H. Wang. A New Human Identification Protocol and Coppersmith's Baby-Step Giant-Step Algorithm. In *ACNS '10*, pages 349–366. Springer-Verlag, 2010.

[3] H. J. Asghar, J. Pieprzyk, and H. Wang. On the Hardness of the Sum of k Mins Problem. *The Computer Journal*, 54(10):1652–1660, Oct. 2011.

[4] R. A. Brualdi. *Introductory Combinatorics*. Pearson Education, Inc., New Jersey, USA, 4th edition, 2004.

[5] J. I. D. Campbell and Q. Xue. Cognitive Arithmetic Across Cultures. *Journal of Experimental Psychology: General*, 130(2):299–315, 2001.

[6] B. Coskun and C. Herley. Can "Something You Know" Be Saved? In *ISC '08*, pages 421–440. Springer-Verlag, 2008.

[7] P. Golle and D. Wagner. Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract). In *SP '07*, pages 66–70. IEEE Computer Society, 2007.

[8] N. J. Hopper and M. Blum. Secure Human Identification Protocols. In *ASIACRYPT '01*, pages 52–66. Springer-Verlag, 2001.

[9] M. Lei, Y. Xiao, S. V. Vrbsky, and C.-C. Li. Virtual Password using Random Linear Functions for On-line Services, ATM Machines, and Pervasive Computing. *Computer Communications*, 31(18):4367–4375, 2008.

[10] M. Lei, Y. Xiao, S. V. Vrbsky, C.-C. Li, and L. Liu. A Virtual Password Scheme to Protect Passwords. In *ICC '08*, pages 1536–1540. IEEE, 2008.

[11] S. Li, H. J. Asghar, J. Pieprzyk, A.-R. Sadeghi, R. Schmitz, and H. Wang. On the Security of PAS (Predicate-based Authentication Service). In *ACSAC '09*, pages 209–218. IEEE Computer Society, 2009.

[12] S. Li, S. A. Khayam, A.-R. Sadeghi, and R. Schmitz. Breaking Randomized Linear Generation Functions based Virtual Password System. In *ICC '10*. IEEE, 2010.

[13] S. Li and H.-Y. Shum. Secure Human-Computer Identification (Interface) Systems against Peeping Attacks: SecHCI. IACR's Cryptology ePrint Archive: Report 2005/268, http://eprint.iacr.org/2005/268, 2005.

[14] X.-Y. Li and S.-H. Teng. Practical Human-Machine Identification over Insecure Channels. *Journal of Combinatorial Optimization*, 3(4):347–361, 1999.

[15] T. Matsumoto. Human-Computer Cryptography: An Attempt. In *CCS '96*, pages 68–75. ACM, 1996.

[16] T. Matsumoto and H. Imai. Human identification through insecure channel. In *EUROCRYPT '91*, pages 409–421. Springer-Verlag, 1991.

[17] T. Perković, A. Mumtaz, Y. Javed, S. Li, S. A. Khayam, and M. Čagalj. Breaking Undercover: Exploiting Design Flaws and Nonuniform Human Behavior. In *SOUPS '11*. ACM, 2011.

[18] S. M. Ross. *A First Course in Probability*. Prentice Hall, 4 edition, 2002.

[19] H. Sasamoto, N. Christin, and E. Hayashi. Undercover: Authentication Usable in Front of Prying Eyes. In *CHI '08*, pages 183–192. ACM, 2008.

[20] L. Sobrado and J.-C. Birget. Graphical Passwords. *The Rutgers Scholar*, 4, 2002.

[21] T.-P. Staff. Debit card skimmers stealing money from accounts, reports say. Times-Picayune, Greater New Orleans, http://www.nola.com/crime/index.ssf/2012/07/debit_card_skimmers_stealing_m.html, 11 July 2012.

[22] S. Sternberg. Memory-scanning: Mental Processes Revealed by Reaction-time Experiments. *American Scientist*, 57(4):421–457, 1969.

[23] C.-H. Wang, T. Hwang, and J.-J. Tsai. On the Matsumoto and Imai's Human Identification Scheme. In *EUROCRYPT '95*, pages 382–392. Springer-Verlag, 1995.

[24] D. Weinshall. Cognitive Authentication Schemes Safe Against Spyware. In *SP '06*, pages 295–300. IEEE Computer Society, 2006.

[25] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme. In *AVI '06*, pages 177–184. ACM, 2006.

[26] X. Bai and W. Gu and S. Chellappan and X. Wang and D. Xuan and B. Ma. PAS: Predicate-Based Authentication Services against Powerful Passive Adversaries. In *ACSAC '08*, pages 433–442. IEEE Computer Society, 2008.

[27] Q. Yan, J. Han, Y. Li, and R. H. Deng. On Limitations of Designing Leakage-Resilient Password Systems: Attacks, Principles and Usability. In *NDSS '12*. Internet Society, 2012.

A. Algorithm to Check if $p(0, m_1) = p(2, m_2)$

Due to the first half-challenge in Foxtail, l has to be greater than or equal to 3. This implies that

$$3n = 2kl \Rightarrow 3n \geq 2k \cdot 3 \Rightarrow k \leq \frac{n}{2}.$$

A further restriction on l, namely $l \leq n$, is imposed by the second-half challenge. The following algorithm finds if $p(0, m_1) = p(2, m_2)$ is true for any values of n, k and l in accordance with these rules. Notice that the expressions of p_i in both $p(0, m_1)$ and $p(0, m_2)$ contain the denominator $\binom{n}{l}$, which can be cancelled on both sides. The algorithm does not compute this factor in the calculation of p_i.

Check $p(0, m_1) \overset{?}{=} p(2, m_2)$**.**

Input: A positive integer n_{max}.

Output: A set of values of the form (n, k, l) satisfying $p(0, m_1) = p(2, m_2)$.

```
initialize found $\leftarrow \emptyset$.
for $n = 1$ to $n_{\mathsf{max}}$ do
    for $k = 1$ to $\lfloor \frac{n}{2} \rfloor$ do
        if $3n \equiv 0 \bmod 2k$ then    # This checks if $3n$ and $2k$ have an integer factor.
```

```
            $l \leftarrow 3n/2k$.
            if $l \leq n$ then
                compute all individual values of $p_i$
                for $p(0, m_1)$ and $p(2, m_2)$.
                if $p(0, m_1) = p(2, m_2)$ then
                    found ← found ∪ $\{(n, k, l)\}$.
output found
```

Our simulations show that the condition is not met for all values of $n_{\mathsf{max}} \leq 5000$.

B. Foxtail with No Window

Since each pass-object appears in the challenge with probability $\frac{1}{2}$, it follows that the probability that a challenge contains c pass-objects is binomially distributed. Hence

$$\tilde{p}_{\mathcal{R}} = \sum_{c \in F_{\mathcal{R}}} \binom{k}{c} \frac{1}{2^k}.$$

To derive an expression for $\xi_\rho^\delta(\mathcal{R}) - \xi_d^\delta(\mathcal{R})$, first note that the probability that a decoy object δ-tuple occurs in a challenge is independent of the number of pass-objects in the challenge. This is true since each object is chosen independently and uniformly at random. So the probability that a decoy object δ-tuple occurs in a challenge is $\frac{1}{2^\delta}$. On the other hand, the probability that a pass-object δ-tuple occurs in a challenge containing c pass-objects is given by

$$\frac{\binom{c}{\delta}}{\binom{k}{\delta}},$$

where $\binom{c}{\delta} = 0$ if $c < \delta$. Combining these two observations, we obtain the following expression in a manner analogous to that of Foxtail with window size $2l$:

$$\xi_\rho^\delta(\mathcal{R}) - \xi_d^\delta(\mathcal{R}) = \frac{1}{\tilde{p}_{\mathcal{R}}} \sum_{c \in F_{\mathcal{R}}} \binom{k}{c} \frac{1}{2^k} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{1}{2^\delta} \right).$$

Now assume $\mathcal{R} = \infty$. Then $\tilde{p}_\infty = 1$, since it is the sum of the probabilities of both responses. We get

$$\begin{aligned} \xi_\rho^\delta(\infty) - \xi_d^\delta(\infty) &= \sum_{c \in F_\infty} \binom{k}{c} \frac{1}{2^k} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{1}{2^\delta} \right) \\ &= \sum_{c=0}^{k} \binom{k}{c} \frac{1}{2^k} \frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \sum_{c=0}^{k} \binom{k}{c} \frac{1}{2^k} \frac{1}{2^\delta} \\ &= \sum_{c=0}^{k} \frac{\binom{k}{c}\binom{c}{\delta}}{\binom{k}{\delta}} \frac{1}{2^k} - \frac{1}{2^\delta} \sum_{c=0}^{k} \binom{k}{c} \frac{1}{2^k}. \end{aligned}$$

Now from [4, §5.8, p. 155], we have

$$\binom{k}{c}\binom{c}{\delta} = \binom{k}{\delta}\binom{k-\delta}{c-\delta}.$$

Also $\sum_{c=0}^{k} \binom{k}{c} \frac{1}{2^k} = 1$, since it is the sum of probabilities of the binomial distribution with parameters $(k, \frac{1}{2})$. We get

$$\begin{aligned} \xi_\rho^\delta(\infty) - \xi_d^\delta(\infty) &= \sum_{c=0}^{k} \frac{\binom{k}{\delta}\binom{k-\delta}{c-\delta}}{\binom{k}{\delta}} \frac{1}{2^k} - \frac{1}{2^\delta} \\ &= \sum_{c=0}^{k} \binom{k-\delta}{c-\delta} \frac{1}{2^k} - \frac{1}{2^\delta} \\ &= \sum_{c=\delta}^{k} \binom{k-\delta}{c-\delta} \frac{1}{2^k} - \frac{1}{2^\delta}. \end{aligned}$$

The last step follows from the fact that $\binom{k-\delta}{c-\delta} = 0$ if $c < \delta$. Let $b = c - \delta$. Then

$$\begin{aligned} \xi_\rho^\delta(\infty) - \xi_d^\delta(\infty) &= \sum_{b=0}^{k-\delta} \binom{k-\delta}{b} \frac{1}{2^k} - \frac{1}{2^\delta} \\ &= \frac{1}{2^\delta} \sum_{b=0}^{k-\delta} \binom{k-\delta}{b} \frac{1}{2^{k-\delta}} - \frac{1}{2^\delta} \\ &= \frac{1}{2^\delta} \cdot 1 - \frac{1}{2^\delta} = 0, \end{aligned}$$

where the term in the next to last step is equal to 1 since it is the sum of probabilities of the binomial distribution with parameters $(k - \delta, \frac{1}{2})$.

C. Why the Flip-bit Fix Prevents RDFA?

To see why the flip-bit fix completely prevents RDFA, we first calculate the probabilities $\tilde{p}_{\mathcal{R}}$. Let p'_b denote the probability that the flip-bit is b. We have

$$\begin{aligned} \tilde{p}_0 &= \sum_{c \in F_0} p_c p'_0 + \sum_{c \in F_1} p_c p'_1 \\ &= \sum_{c \in F_0} p_c p'_1 + \sum_{c \in F_1} p_c p'_0 \\ &= \tilde{p}_1, \end{aligned}$$

where we have used the fact that $p'_0 = p'_1$. Let $q(c)$ and $q'(c)$ denote the probabilities that a pass-object and a decoy object δ-tuple is present in a challenge containing c objects, respectively. Then

$$q_c = \frac{\binom{c}{\delta}}{\binom{k}{\delta}}, \; q'_c = \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}}.$$

We have

$$\begin{aligned}
\xi_\rho^\delta(0) &= \frac{1}{\tilde{p}_0}\left(\sum_{c\in F_0} p_c p'_0 q_c + \sum_{c\in F_1} p_c p'_1 q_c\right) \\
&= \frac{1}{\tilde{p}_0}\left(\sum_{c\in F_0} p_c p'_1 q_c + \sum_{c\in F_1} p_c p'_0 q_c\right) \\
&= \frac{1}{\tilde{p}_1}\left(\sum_{c\in F_0} p_c p'_1 q_c + \sum_{c\in F_1} p_c p'_0 q_c\right) \\
&= \xi_\rho^\delta(1).
\end{aligned}$$

Let $\overline{\mathcal{R}}$ denote the compliment of $\mathcal{R}$. Then for both values of $\mathcal{R}$

$$\begin{aligned}
\xi_\rho^\delta(\mathcal{R}) &= \frac{1}{\tilde{p}_\mathcal{R}}\left(\sum_{c\in J_\mathcal{R}} p_c p'_\mathcal{R} q_c + \sum_{c\in J_{\overline{\mathcal{R}}}} p_c p'_{\overline{\mathcal{R}}} q_c\right) \\
&= \frac{1}{\tilde{p}_\mathcal{R}} p'_\mathcal{R} \sum_{c=0}^{k} p_c q_c \\
&= \frac{1}{\frac{1}{2}}\frac{1}{2} \sum_{c=0}^{\min\{k,l\}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \frac{\binom{c}{\delta}}{\binom{k}{\delta}} \\
&= \sum_{c=0}^{\min\{k,l\}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \frac{\binom{c}{\delta}}{\binom{k}{\delta}},
\end{aligned}$$

where we have used the fact that $\tilde{p}_\mathcal{R} = \frac{1}{2}$ and $p'_\mathcal{R} = p'_{\overline{\mathcal{R}}} = \frac{1}{2}$. Similarly for the decoy objects we get $\xi_d^\delta(0) = \xi_d^\delta(1)$ and

$$\xi_d^\delta(\mathcal{R}) = \sum_{c=0}^{\min\{k,l\}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}}.$$

Combining these two results, we get

$$\xi_\rho^\delta(\mathcal{R}) - \xi_d^\delta(\mathcal{R}) = \sum_{c=0}^{\min\{k,l\}} \frac{\binom{k}{c}\binom{n-k}{l-c}}{\binom{n}{l}} \left(\frac{\binom{c}{\delta}}{\binom{k}{\delta}} - \frac{\binom{l-c}{\delta}}{\binom{n-k}{\delta}}\right).$$

This is the same equation as that of the no-response differential in the case of the Rand-rule Foxtail as shown in Table 2, which leads to the conclusion that $\xi_\rho^\delta(\mathcal{R}) - \xi_d^\delta(\mathcal{R}) = 0$, for all $\mathcal{R}$ and δ.