

Neuroscience Meets Cryptography: Designing Crypto Primitives Secure Against Rubber Hose Attacks

Hristo Bojinov Daniel Sanchez, Paul Reber Dan Boneh Patrick Lincoln
Stanford University Northwestern University Stanford University SRI

Abstract

Cryptographic systems often rely on the secrecy of cryptographic keys given to users. Many schemes, however, cannot resist coercion attacks where the user is forcibly asked by an attacker to reveal the key. These attacks, known as *rubber hose cryptanalysis*, are often the easiest way to defeat cryptography. We present a defense against coercion attacks using the concept of *implicit learning* from cognitive psychology. Implicit learning refers to learning of patterns without any conscious knowledge of the learned pattern. We use a carefully crafted computer game to plant a secret password in the participant's brain without the participant having any conscious knowledge of the trained password. While the planted secret can be used for authentication, the participant cannot be coerced into revealing it since he or she has no conscious knowledge of it. We performed a number of user studies using Amazon's Mechanical Turk to verify that participants can successfully re-authenticate over time and that they are unable to reconstruct or even recognize short fragments of the planted secret.

1 Introduction

Consider the following scenario: a high security facility employs a sophisticated authentication system to check that only persons who know a secret key, possess a hardware token, and have an authorized biometric can enter. Guards ensure that only people who successfully authenticate can enter the facility. Now, suppose a clever attacker captures an authenticated user. The attacker can steal the user's hardware token, fake the user's biometrics, and coerce the victim into revealing his or her secret key. At this point the attacker can impersonate the victim and defeat the expensive authentication system deployed at the facility.

So-called rubber hose attacks have long been the bane of security systems and are often the easiest way to de-

feat cryptography [22]. The problem is that an authenticated user must possess authentication credentials and these credentials can be extracted by force [19] or by other means.

In this work we present a new approach to preventing rubber hose attacks using the concept of *implicit learning* [5, 17] from cognitive psychology. Implicit learning is believed to involve the part of the brain called the *basal ganglia* that learns tasks such as riding a bicycle or playing golf by repeatedly performing those tasks. Experiments designed to trigger implicit learning show that knowledge learned this way is not consciously accessible to the person being trained [17]. An everyday example of this phenomenon is riding a bicycle: we know how to ride a bicycle, but cannot explain how we do it. Section 2 gives more background of the relevant neuroscience.

Implicit learning presents a fascinating tool for designing coercion-resistant security systems. In this paper we focus on user authentication where implicit learning is used to plant a password in the human brain that can be detected during authentication, but cannot be explicitly described by the user. Such a system avoids the problem that people can be persuaded to reveal their password. To use this system, participants would be initially trained to do a specific task called Serial Interception Sequence Learning (SISL), described in the next section. Training is done using a computer game that results in implicit learning of a specific sequence of key strokes that functions as an authentication password. In our experiments, training sessions last approximately 30 to 45 minutes and participants learn a random password that has about 38 bits of entropy. We conducted experiments to show that after training, participants cannot reconstruct the trained sequence and cannot even recognize short fragments of it.

To be authenticated at a later time, a participant is presented with multiple SISL tasks where one of the tasks contains elements from the trained sequence. By exhibiting reliably better performance on the trained ele-

ments compared to untrained, the participant validates his or her identity within 5 to 6 minutes. An attacker who does not know the trained sequence cannot exhibit the user’s performance characteristics measured at the end of training. Note that the authentication procedure is an interactive game in which the server knows the participant’s secret training sequence and uses it to authenticate the participant. Readers who want to play with the system can check out the training game at brainauth.com/testdrive.

While in this paper we focus on coercion-resistant user authentication systems, authentication is just the tip of the iceberg. We expect that many other coercion-resistant security primitives can be designed using implicit learning.

Threat model. The proposed system is designed to be used as a local password mechanism requiring physical presence. That is, we consider authentication at the entrance to a secure location where a guard can ensure that a real person is taking the test without the aid of any electronics.

To fool the authentication test the adversary is allowed to intercept one or more trained users and get them to reveal as much as they can, possibly using coercion. Then the adversary, on his own, engages in the live authentication test and his goal is to pass the test.

We stress that as with standard password authentication, the system is not designed to resist eavesdropping attacks such as shoulder surfing during the authentication process. While challenge-response protocols are a standard defense against eavesdropping, it is currently an open problem to design a challenge-response protocol based on implicit learning. We come back to this question at the end of the paper.

Benefits over biometric authentication. The trained secret sequence can be thought of as a biometric key authenticating the trained participant. However, unlike biometric keys the authenticating information cannot be surreptitiously duplicated and participants cannot reveal the trained secret even if they want to. In addition, if the trained sequence is compromised, a new identifying sequence can be trained as a replacement, resulting in a change of password.

We discuss other related work in Section 6, but briefly mention here a related result of Denning et al. [4] that uses images to train users to implicitly memorize passwords. This approach is not as resistant to rubber hose attacks since users will remember images they have seen versus ones they have not, giving an attacker information that can be used for authentication. Additionally, image-based methods require large sets of images to be prepared and used only once per user making the system difficult to deploy. Our combinatorial approach lets us

lower bound the entropy of the learned secrets, is simple to set up, and is designed to leave no conscious trace of the trained sequences.

User studies. To validate our proposal we performed a number of user studies using Amazon’s Mechanical Turk. We asked the following core questions that explore the feasibility of authentication via implicit learning:

- Is individual identification reliable? That is, can trained users re-authenticate and can they do it over time?
- Can an attacker reverse engineer the sequence from easily obtained performance data from a trained participant?

Across three experiments, we present promising initial results supporting the practical implementation of our design. First, we show that identification is possible with relatively short training and a simple test. Second, the information learned by the user persists over delays of one and two weeks: while there is some forgetting over a week, there is little additional forgetting at two weeks suggesting a long (exponentially shaped) forgetting curve. Finally, in a third experiment we examined an attack based on having participants complete sequences containing all minimal-length fragments needed to try to reconstruct the identification sequence: our results show that participants do not express reliable sequence knowledge under this condition, indicating that the underlying sequence information is resistant to attack until longer subsequences are guessed correctly by the attacker.

2 An Overview of the Human Memory System

The difference between knowing how to perform a well-learned skill and being able to explain that performance is familiar to anyone who has acquired skilled expertise. This dissociation reflects the multiple memory systems in the human brain [14]. Memory for verbally reportable facts, events and episodes depends on the medial temporal lobe memory system (including the hippocampus). Damage to this system due to stroke, Alzheimer’s disease neuropathology, or aging leads to impairments in conscious, explicit memory. However, patients with impairments to explicit memory often show an intact ability to acquire new information implicitly, including exhibiting normal learning of several kinds of skills. The types of learning preserved in memory-disordered patients are those learned incidentally through practice: even in healthy participants the information thus acquired cannot be easily verbally described.

Several decades of experimental cognitive psychology have led to the development of tasks that selectively de-

pend on this type of implicit, non-conscious learning system. These tasks typically present information covertly with embedded structure in a set of experimental stimuli. Although participants are not attempting to learn this structure, evidence for learning can be observed in their performance.

The covertly embedded information often takes the form of a statistical structure to a sequence of responses. Participants exhibit improved performance when the responses follow this sequence and performance declines if the structure is changed [12]. The improvement in performance can occur completely outside of awareness, that is, participants do not realize there is any structure nor can they recognize the structure when shown [17]. The lack of awareness of learning indicates the memory system supporting learning is not part of the explicit, declarative memory system and instead is hypothesized to depend on the basal ganglia and connections to motor cortical areas [6].

Less is known about the information processing characteristics of the cortico-striatal memory system operating in the connections between the basal ganglia and motor cortical areas. Most prior research has examined learning of simple structures with small amounts of information, typically repeating sequences of actions 10-12 items in length. However, more recent studies have found that long, complex sequences can be learned fairly rapidly by this memory system and that learning is relatively unaffected by noise [18]. The ability to learn repeating sequences that are at least 80 items long relatively rapidly and the fact that this training can be hidden within irrelevant responses (noise) during training suggests an intriguing possibility for covertly embedding non-reportable cryptographic data within the cortico-striatal memory system in the human brain.

2.1 The SISL Task and Applet

The execution of the Serial Interception Sequence Learning (SISL) task is central to the authentication system that we have developed. Here we introduce the SISL task in the context of the human memory system in order to provide background for describing our design and practical experiments.

Originally introduced in [17], SISL is a task in which human participants develop sensitivity to structured information without being aware of what they have learned. The task requires participants to intercept moving objects (circles) delivered in a pre-determined sequence, much like this is done in the popular game “Guitar Hero”. Initially each object appears at the top of one of four different columns, and falls vertically at a constant speed until it reaches the “sink” at the bottom, at which point it disappears. The goal for the player is to

intercept every object as it nears the sink. Interception is performed by pressing the key that corresponds to the object’s column when the object is in the correct vertical position. Pressing the wrong key or not pressing any key results in an incorrect outcome for that object. In a typical training session of 30-60 minutes, participants complete several thousand trials and the order of the cues follows a covertly embedded repeating sequence on 80% of trials. The game is designed to keep each user at (but not beyond) the limit of his or her abilities by gradually varying the speed of the falling circles to achieve a hit rate of about 70%. Knowledge of the embedded repeating sequence is assessed by comparing the performance rate (percent correct) during times when the cues follow the trained sequence to that during periods when the cues follow an untrained sequence.

All of the sequences presented to the user are designed to prevent conspicuous, easy to remember patterns from emerging. Specifically, training as well as random sequences are designed to contain every ordered pair of characters exactly once with no character appearing twice in a row, and thus the sequence length must be $4 \times 3 = 12$ when four columns (characters) are used. The result is that while the trained sequence is performed better than an untrained sequence, the participant usually does not consciously recognize the trained sequence. In order to confirm this in experimental work, after SISL participants are typically asked to complete tests of explicit recognition in which they specify how familiar various sequences look to them.

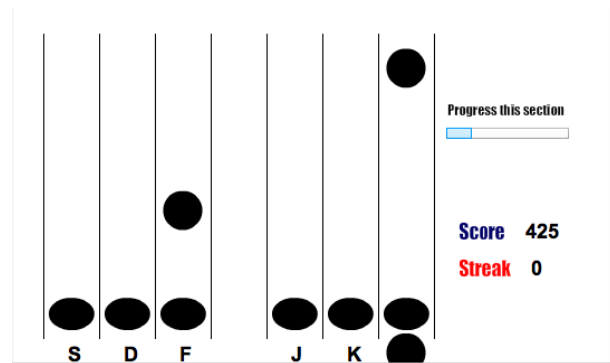


Figure 1: Screenshot of the SISL task in progress.

For the current application, we extended the traditional definition of the SISL task in order to accommodate its use as an authentication mechanism. First, we increased the number of columns to six, which increases the potential complexity of the trained sequence. Using the same constraints on sequence order as the 4-column version of the task, the training sequences are 30 items long. As a result, the number of possible sequences that can be used

as a secret key is increased exponentially from only 256 to nearly 248 billion, as explained in the next section. Second, we added an empty column in the middle of the layout (Figure 1). In early experimental testing we found out that the empty column facilitates the visual perception of the falling objects and helps the user to “map” them to the correct hand, especially for objects in the middle columns which are otherwise easily confused at high speed.

The SISL task is delivered to users as a Flash application via a web browser. Participants navigate to our web site, www.brainauth.com, and are presented with a consent form. Once they agree to participate, the applet downloads a random training sequence and starts the game. Upon completion of the training and test trials, the explicit recognition test is administered, and results are uploaded to the server. Once we describe our authentication system, we will return to describe how the SISL applet functions in the bigger scheme of our experiments with multiple users.

3 The Basic Authentication System Using Implicit Learning

The SISL task provides a method for storing a secret key within the human brain that can be detected during authentication, but cannot be explicitly described by the user. Such a system avoids the problem that people can be persuaded to reveal their password and can form the basis of a coercion-resistant authentication protocol. If the information is compromised, a new identifying sequence can be trained as a replacement—resulting in a change of password.

The identification system operates in two steps: training followed by authentication. In the training phase, the secret key learned by the user is as in the expanded SISL task, namely a sequence of 30 characters over the set $S = \{s, d, f, j, k, l\}$. We only use 30-character sequences that correspond to an Euler cycle in the graph shown in Figure 2 (i.e. a cycle where every edge appears exactly once). These sequences have the property that every non-repeating bigram over S (such as ‘sd’, ‘dj’, ‘fk’) appears exactly once. In order to anticipate the next item (e.g., to show a performance advantage), it is necessary to learn associations among groups of three or more items. This eliminates learning of letter frequencies or common pairs of letters, which reduces *conscious* recognition of the embedded repeating sequence [5].

Let Σ denote the set of all possible secret keys, namely the set of 30-character sequences corresponding to Euler cycles in Figure 2. The number of Euler cycles in this graph can be computed using the BEST theorem [20]

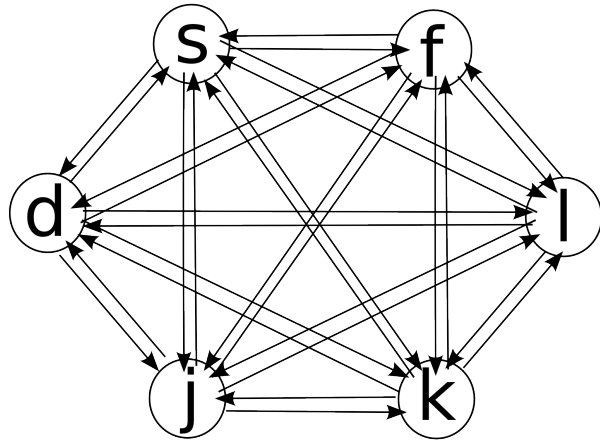


Figure 2: The secret key we generate is a random 30-character sequence from the set of Euler cycles in this directed graph. The resulting sequence contains all bigrams exactly once, excluding repeating characters.

which gives

$$\#\text{keys} = 6^4 \cdot 24^6 \approx 2^{37.8}.$$

Hence the learned random secret has about 38 bits of entropy which is far more than the entropy of standard memorized passwords.

Training. Users learn a random 30-item secret key $k \in \Sigma$ by playing the SISL game in a trusted environment. To train users we experimented with the following procedure:

- While performing the SISL task the trainee is presented with the 30-item secret key sequence repeated three times followed by 18 items selected from a random other sequence (subject to the constraint that there will be no back-to-back repetitions of the same cue), for a total of 108 items.
- This sequence is repeated five times, so that the trainee is presented with a total of 540 items.
- At the end of this sequence there is a short pause in the SISL game and then the entire sequence of 540 items (including the pause at the end) is repeated six more times.

During the entire training session the trainee is presented with $7 \times 540 = 3780$ items which takes approximately 30-45 minutes to complete. After the training phase completes, the trainee runs through the authentication test described next to ensure that training succeeded. The system records the final playing speed that the user achieved.

SISL Authentication. To authenticate at a later time, a trained user is presented with the SISL game where the structure of the cues contains elements from the trained authentication sequence and untrained elements for comparison. By exhibiting reliably better performance on the trained elements compared to untrained, the participant validates his or her identity. Specifically we experimented with the following authentication procedure:

- Let k_0 be the trained 30-item sequence and let k_1, k_2 be two additional 30-item sequences chosen at random from Σ . The same sequences (k_0, k_1, k_2) are used for all authentication sessions.
- The system chooses a random permutation π of $(0, 1, 2, 0, 1, 2)$ (e.g., $\pi = (2, 1, 0, 0, 2, 1)$) and presents the user with a SISL game with the following sequence of $540 = 18 \times 30$ items:

$$k_{\pi_1}, k_{\pi_1}, k_{\pi_1}, \dots, k_{\pi_6}, k_{\pi_6}, k_{\pi_6}.$$

That is, each of k_0, k_1, k_2 is shown to the user exactly six times (two groups of three repetitions), but ordering is random. The game begins at the speed at which the training for that user ended.

- For $i = 0, 1, 2$ let p_i be the fraction of correct keys the user entered during all plays of the sequence k_i . The system declares that authentication succeeded if

$$p_0 > \text{average}(p_1, p_2) + \sigma \quad (3.1)$$

Where $\sigma > 0$ is sufficiently large to minimize the possibility that this gap occurred by chance, but without causing authentication failures.

In the above, preliminary formulation, the authentication process is potentially vulnerable to an attack by which an untrained user degrades his performance across two blocks hoping to exhibit an artificial performance difference in favor of the trained sequence (and obtaining a 1/3 chance of passing authentication). We discuss a robust defense against this in Section 5, but for now we mention that two simple precautions offer some protection, even for this simple assessment procedure. First, verifying that the authenticator is a live human makes it difficult to consistently change performance across the foil blocks k_1, k_2 . Second, the final training speed obtained during acquisition of the sequence is known to the authentication server and the attacker is unlikely to match that performance difference between the trained and foil blocks. A performance gap that is substantially different from the one obtained after training indicates an attack.

Analysis. The next two sections discuss two critical aspects of this system:

- Usability: can a trained user complete the authentication task reliably over time?
- Security: can an attacker who intercepts a trained user coerce enough information out of the user to properly authenticate?

4 Usability Experiments

We report on preliminary experiments that demonstrate feasibility and promise of the SISL authentication system. We carried out the experiments in three stages. First, we established that reliable learning was observed with the new expanded version of the SISL task using Mechanical Turk. Second, we verified that users retain the knowledge of the trained sequence after delays of one and two weeks. Finally, we investigated the effectiveness of an attack on participants' sequence knowledge based on sampling the smallest fragments from which the original sequence could potentially be reconstructed.

The experiments were carried out online within Amazon's Mechanical Turk platform. The advantages of Mechanical Turk involve a practically unlimited base of participants, and a relatively low cost. One drawback of running the experiments online is the relative lack of control we had over users coming back at a later time for repeat evaluations. We discuss all of these considerations towards the end of the section.

4.1 Experiment 1: Implicit and Explicit Learning

Our first experiment confirmed that implicit learning can be clearly detected while explicit conscious sequence knowledge was minimal. Experimental data from 35 participants were included in the analysis.

The experiment used the training procedure described in the previous section where the training phase contained 3780 total trials and took approximately 30-45 minutes to complete. Recall that training consists of seven 540-trial training blocks. After the training session, participants completed a SISL authentication test that compares performance on the trained sequence to performance on two random test sequences.

Learning of the trained sequence is shown in Figure 3 as a function of the performance advantage (increase in percent correct responses) for the trained sequence compared with the randomly occurring noise segments. On the test block following training, participants performed the SISL task at an average rate of 79.2% correct for the trained sequence and 70.6% correct for the untrained sequence. The difference of 8.6% correct (SE 2.4%)¹

¹SE is short hand for *Standard Error*.

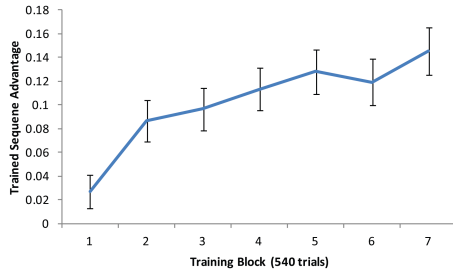


Figure 3: Across training participants gradually begin to express knowledge of the repeating sequence by exhibiting a performance advantage for the trained sequence compared to randomly interspersed noise segments. Note that overall performance on the task stays at around 70% throughout due to the adaptive nature of the task by which the speed is increased as participants become better at general SISL performance.

indicated reliably better performance for the trained sequence. By one-sample *t*-test versus zero, the expected difference between trained and untrained if there was no learning² would be $t(34) = 3.55$, $p < .01$.

Group-level differences in performance are commonly seen on tests of implicit learning, but being able to reliably assess individual learning is necessary for an authentication method. On an individual participant basis, performance on the trained sequence could be discriminated from the untrained sequence on the 540 test trials (by chi-squared analysis at $p < .05$) in 25 of 35 cases. For authentication purposes, the individual reliability of the assessment will need to be further improved by longer training to establish the implicitly learned sequence. However, the ability to identify learning in a large fraction of individuals with relatively short training is a feature of the SISL task not seen in most tests of implicit learning.

Explicit recognition test. After the training and test blocks, participants were presented with five different animated sequences and asked how familiar each looked on a scale of 0 to 10). Of the five sequences, one was the trained sequence and the other four were randomly selected foils. This test assessed explicit recognition memory for the trained sequence.

On the recognition test, participants rated the trained

²In other words, if the percent correct measurements for trained and untrained sequences followed the same normal distribution, the *t*-value calculated with $N = 35$ samples (and thus $N - 1 = 34$ degrees of freedom), should be near zero—less than 3.55 with 99% probability ($p = 0.01$); in contrast, the value we obtained was 8.6. The *t*-test is a standard statistical method used to confirm that the manipulated variable (here, sequence type) affects the measured variable (performance correct).

sequence as familiar at an average of 6.5 (SE 0.4) on the 0-10 scale and rated novel untrained sequences at 5.15 (SE 0.3). The modestly higher recognition of the trained sequence was reliable across the group, $t(34) = 3.69$, $p < .01$, but did not correlate with SISL performance ($r = 0.13$) indicating that it did not contribute to the implicit test. Slightly higher recognition of the trained sequence is often seen in implicit learning experiments as healthy participants find some parts of the training sequence familiar after practice. It is worth noting that implicit memory does not transform into explicit knowledge, even with repeated use, and the structure and length of the training and test sequences specifically aim to reduce the possibility that explicit knowledge is accumulated over time.

The general small difference in recognition ratings (5.15 vs. 6.5) indicates that participants would not be able to recall the 30-item sequence meaning that they could not consciously produce the training information (e.g. to compromise the security of the authentication method). One participant remarked in a follow-up email message:

“... To be honest I was not that sure of the quizzes at the end. When I played the tempo was so high it was incredibly difficult to keep a track of the circles. Most of the time my fingers moved by themselves, at least it felt that way. I noticed two repeating patterns over all the levels. (I’m not totally sure what the buttons were, was it DFG JKL?) One was D-F-G-F-D I think and the other I’m not quite sure the sequence but it was a four or five button series which went from the left to the right and back to the left...”

We discuss the reconstruction question further in our third experiment.

4.2 Experiment 2: Recall Over Time

An authentication mechanism is only useful if authentication can still be accurately performed at some time after the password is memorized. In Experiment 2, we confirmed that sequence-specific knowledge acquired by users was retained over prolonged periods of time. Although skill learning generally persists over time, a SISL-based test had never been conducted with a substantial delay and a sufficient number of participants.

In Experiment 2, participants agreed to perform the SISL task over two sessions. In the first session, participants completed a training sequence which the same structure as the one in Experiment 1. The training was immediately followed by the same SISL test to assess sequence knowledge before the delay. A group of 32 participants returned to the online applet after 1 week to

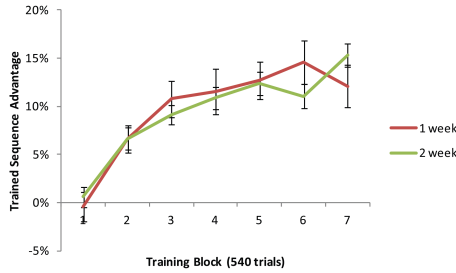


Figure 4: Across training participants gradually begin to express knowledge of the repeating sequence by exhibiting a performance advantage for the trained sequence compared to randomly interspersed noise segments. Learning performance was similar across both groups and similar to Experiment 1, as expected.

perform a retention test and recognition assessment for the trained sequence. A separate group of 80 participants returned after a 2 week delay for the retention and recognition tests. For the 1-week group, the test session consisted of a 540-trial implicit sequence learning assessment. For the 2-week group, the test session was doubled in length to additionally evaluate whether a longer test provided better sensitivity to individual sequence knowledge. For both groups, the initial speed of the test on the delay session was set to match the speed with which the participants had been performing the task at the end of the training session. A short warm-up block of 180 trials was used to adjust this initial speed so that participants were performing at around the target 70% correct at the beginning of the retention test.

Figure 4 shows gradual learning of the trained sequence during the first session for both groups as in Experiment 1. Implicit sequence knowledge at both immediate and delayed tests is shown in Figure 5. On all five assessments, participants exhibited reliable sequence learning as a group, $t_s > 4.3$, $p_s < .01$. On the one-week delay test, 15 of 32 participants exhibited individually reliable sequence knowledge. However, for the two-week delay group, 49 of 80 participants exhibited reliable sequence knowledge reflecting the increased sensitivity in the longer assessment test used. Future research will examine both increased training time and assessment tests with increased sensitivity to individual knowledge to provide a reliable and accurate identification method by SISL performance.

Even at one and two weeks delay, participants exhibited the same modest tendency for better recognition of the trained sequence, $t_s > 2.8$, $p_s < .05$. Again, recognition performance did not correlate with expression of sequence knowledge, $r_s < .16$ and did not suggest any

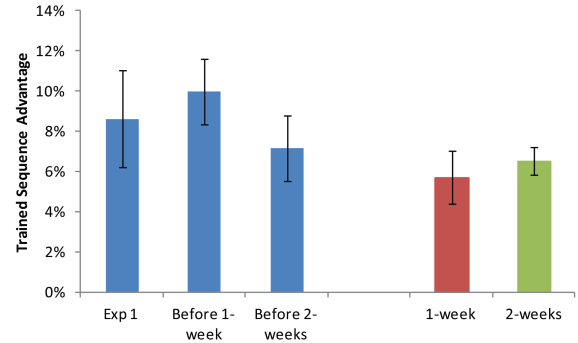


Figure 5: Participants exhibited reliable sequence knowledge on both immediate assessments (shown for Experiment 1 and both conditions of Experiment 2) shown by a performance advantage for the trained sequence compared with untrained, novel sequences at test. Sequence knowledge is retained at both the 1 and 2 week delay test sessions. While there is some reduction in expressed knowledge after either delay, the lack of significant additional decay from 1 to 2 weeks suggests that information is likely to persist for significant periods following 2 weeks (exponential or power-law decay curves are commonly observed for many types of memory).

ability to recall the entire 30-item trained sequences.

4.3 Mechanical Turk

Running our experiments over Mechanical Turk required considerable thought and effort to ensure that the experiments do not suffer from selection bias and are conducted fairly for both participants and researchers.

One of the early initial questions was that of setting the price for user participation. The training block, which comprises the bulk of the initial session, takes approximately 30-40 minutes to complete depending on player skill. We wanted to motivate our participants to perform to the best of their abilities, and thus set a price of \$5.00 for standalone sessions, assuming a total of approximately one hour of work involved. Apart from isolated complaints from users who thought the game moved too slowly (likely due to them not pressing keys, or playing incorrectly), most users were happy to participate and even solicited additional work. We defined our HIT (Human Intelligence Task) such that each worker could participate only once in it and we believe that there were few—if any—cases where the same user submitted multiple responses.

We had to design special incentives for participants to return and complete the second part in the case of two-session experiments. The approach that worked well for

us was to price the initial (much lengthier) part at \$2.00 and the follow-up 15-minute session at \$6.00. We also explained clearly that this is a two-HIT sequence, and that payment for both parts will only be processed once the second part is done. No-shows at the second session would get no payment at all. Additionally we used Amazon’s command line tools to automatically send reminders to participants when the second session was available and due. As a result, we saw over 90% of the people who completed the first session return and finish the second part.

Due to the special requirements of the SISL application we had to create what is considered to be an “external HIT”, exposing the task as a public website. In order to make sure that results submitted in Amazon correspond to valid submissions in our system, we designed a system that involves a receipt code for every successfully completed session. The code is a 6-digit number between 100000 and 999999—we chose this size to prevent people from easily guessing the code, but not make it difficult for them to write it down (especially useful in two-session experiments, where we also have to fetch the correct follow-up sequence that matches the user’s first visit). After follow-up sessions we provided the user with a second code that needed to be submitted to the separate second HIT in order to receive payment.

Naturally we were concerned about the security of our system, so we took measures to only accept limited types of input as parameters, leaving the website open mostly to denial of service attacks which we had no reason to expect. In comparison, our fear of legitimate users trying to cheat the system and getting paid without completing quality work was somewhat more justified. We saw some limited instances of behavior in this category:

- There were users who, against the instructions, submitted an invalid receipt code. We immediately rejected any such submissions.
- Some users submitted sequences that were so long that they did not fit in our generous allowance on the server. Upon examination we found out that these were due primarily to excessive wrong key presses (sometimes 5 or more key presses for the same object, which suggests that possibly an automated tool was used to complete the task).
- In relatively few situations we noticed users who had unusually long intervals of inactivity. We excluded the most outrageous submissions but leaned towards including the rest in the results of the study in order to avoid biasing our data towards people who did well.

The scope of these abuses never amounted to more than 5% of the submissions, and we believe that the

Experiment	Part	Submissions		
		All	Paid	Used
baseline		46	39	34
1 week delay	initial	35	32	32
1 week delay	follow-up	45	32	32
2 week delay	initial	100	95 (a)	82
2 week delay	follow-up	111	84 (b)	82
trigrams		37	34	32

Table 1: Total number of participants in each experiment. The higher number of submissions on follow-up session are due to more failed opportunistic attempts by users to get paid \$6.00 for no work because HIT assignments were remaining available longer, waiting for eligible users to show up. Notes: (a) we paid more people than necessary due to the 16-day auto-approval configuration of the HIT; (b) we paid, but did not evaluate a submission which came in after the cut-off time; (c) the variation in number of participants across experiments was due to varying response and acceptance rates—our primary goal was to collect enough data to be able to make statistical inferences, and we deliberately collected more data for the most difficult experiment (the 2-week delay).

organization of the Mechanical Turk system is at least partially to thank: workers need to register, and provide some sort of payment account which makes their identity relatively easy to track; moreover, rejected work negatively affects a worker’s score and as a result most users genuinely try to do the best they can, get entertained if possible, and earn some extra money in the process. Overall, we consider our use of Mechanical Turk to have been a big success: it allowed us to conduct each experiment practically overnight, drawing on the huge available pool of participants.

5 Security Analysis

In this section we analyze the security of the basic authentication protocol from Section 3 and propose a number of extensions that improve security. We also experiment with a particular attack that attempts to extract the secret sequence from the user one fragment at a time. Our Mechanical Turk experiment shows that this attack works poorly on humans.

5.1 Implicit Learning as a Cryptographic Primitive

We begin with an abstract model of the new functionality enabled by implicit learning. Traditional modeling

of participants in a cryptographic protocol are as entities who hold secrets unknown to the adversary. These assumptions fall apart in the face of coercion since all secrets can be extracted from the participant.

Implicit learning provides the following new abstract functionality: the training phase embeds a predicate

$$p : \Sigma \rightarrow \{0, 1\}$$

in the user’s brain for some large set Σ . Anyone can ask the user to evaluate his or her predicate p at a point $k \in \Sigma$. The predicate evaluates to 1 when k has been learned by the user and evaluates to 0 otherwise. The number of inputs at which p evaluates to 1 is relatively small. Most often p will only evaluate to 1 at a single point meaning that the user has been trained on only one secret sequence.

The key feature of implicit learning is that even under duress it is impossible to extract a point $k \in \Sigma$ from the user for which $p(k) = 1$. This abstract property captures the fact that the secret sequence k is implicitly learned by the user and not consciously accessible. In this paper, we use the implicit learning primitive to construct an authentication system, but one can imagine it being used more broadly in security systems.

The authentication procedure described in Section 3 provides an implementation of the predicate $p(\cdot)$ for some sequence k_0 in Σ . If the procedure declares success we say that $p(k_0) = 1$ and otherwise $p(k_0) = 0$. The predicate p is embedded in the user’s brain during the training session.

The basic coercion threat model. The SISL authentication system from Section 3 is designed to resist an adversary who tries to fool the authentication test. We assume the test requires physical presence and begins with a liveness check to ensure that a real person is taking the test without the aid of any instruments. To fool the authentication test the adversary is allowed the following sequence of steps:

- Extraction phase: intercept one or more trained users and get them to reveal as much as they can, possibly using coercion.
- Test phase: the adversary, on his own, submits to the authentication test and his or her goal is to pass the test. In real life this could mean that the adversary shows up at the entrance to a secure facility and attempts to pass the authentication test there. If he fails he could be detained for questioning.

This basic threat model gives the attacker a single chance at the authentication test. We consider a model where the attacker may iterate the extraction and test

phases, alternating between extraction and testing, later on in this section.

We also note that the basic threat model assumes that during the training phase, when users are taught the credential, users are following the instructions and are not deliberately trying to mislead the training process. In effect, the adversary is only allowed to coerce a user after the training process completes.

It is straight-forward to show that the system of Section 3 is secure under this basic threat model, assuming the training procedure embeds an implicitly learned predicate p in the user’s brain. Indeed, if the attacker intercepts u trained users and subjects each one to q queries, his chances of finding a valid sequence is at most $qu/|\Sigma|$. Since each test takes about five minutes, we can assume an upper bound of $q = 10^5$ trials per captured user (this amounts to about one year of non-stop testing per user which will either interfere with the user’s learned password rendering the user useless to the attacker, or alert security administrators due to the user’s absence prompting a revocation of the credentials). Hence, even after capturing $u = 100$ users, the attacker’s success probability is only

$$100 \times 10^5 / |\Sigma| \approx 2^{-16}.$$

Further complicating the attacker’s life is the fact that subjecting a person to many random SISL games may obliterate the learned sequence or cause the person to learn an incorrect sequence thereby making extraction impossible.

We note that physical presence is necessary in authentication systems designed to resist coercion attacks. If the system supported remote authentication then an attacker could coerce a trained user to authenticate to a remote server and then hijack the session.

Security enhancements. The security model above gives the attacker one chance to authenticate and the attacker must succeed with non-negligible probability. If the attacker is allowed multiple authentication attempts — iterating the extraction and test phases, alternating between the two — then the protocol may become insecure. The reason is that during an authentication attempt the attacker sees the three sequences k_0, k_1, k_2 and could memorize one of them (30 symbols). He would then train offline on that sequence so that at the next authentication attempt he would have a 1/3 chance in succeeding. If the attacker could memorize all three sequences (90 symbols), he could offline subject a trained user to all three sequences and reliably determine which is the correct one and then train himself on that sequence. He is then *guaranteed* success at the next authentication trial. We note that this attack is non-trivial to pull off since

it can be difficult for a human attacker to memorize an entire sequence at the speed the game is played.

Another potential attack, already discussed in Section 3, is an attacker who happens to be an expert player, but deliberately degrades his performance on two of the sequences presented. With probability $1/3$ he will show a performance gap on the correct sequence and pass the authentication test. We described a number of defenses in Section 3. Here we describe a more robust defense.

Both attacks above can be defeated with combinatorics. Instead of training the user on a single sequence, we train the user on a small number of sequences, say four. Experiments [18] suggest that the human brain can learn multiple sequences and these learned sequences do not interfere with one another. Equivalently we could train the user on a longer sequence and use its fragments during authentication. While this will increase training time, we show that it can enhance security.

During authentication, instead of using one correct sequence and two foils, we use the four correct sequences randomly interspersed within 8 foils. Authentication succeeds if the attacker shows a measurable performance gap on the correct 4 out of 12 presented sequences. An attacker who slows down on random sequences will now have at most a $1/\binom{12}{4} \approx 1/500$ chance in passing the test. The number of trained sequences (4) and the number of foils (8) can be adjusted to achieve an acceptable tradeoff between security and usability.

Similarly, a small number of authentication attempts will not help a direct attacker pass the test. However, memorizing the authentication test (360 symbols) and later presenting it to a coerced user could give the adversary an advantage. To further defend against this memorization attack we add one more step to the authentication procedure: once the authentication server observes that the user failed to demonstrate a measurable gap on some of the trained sequences, all remaining trained sequences are replaced with random foils. This ensures that an attacker who tries to authenticate with no prior knowledge will not see all the trained sequences and therefore cannot extract all trained sequences from a coerced user. Consequently, a *one-shot* attack on a coerced user is not possible. Nevertheless, by iterating this process — taking the authentication test, memorizing the observed sequences, and then testing them out on a coerced trained user — the attacker may eventually learn all trained sequences and succeed in fooling the authentication test. During this process, however, the attacker must engage in the authentication test where he demonstrates knowledge of a strict subset of the trained sequences, but cannot demonstrate knowledge of all sequences. This is a clear signal to the system that it is under attack at which point the person engaging in the authentication test could

be detained for questioning and the legitimate user is blocked from authenticating with the system until he or she is retrained on a new set of sequences.

Eavesdropping security. Traditional password authentication is vulnerable to eavesdropping (either via client-side malware or shoulder surfing) and so is the authentication system presented here. An eavesdropper who obtains a number of valid authentication transcripts with a trained user will be able to reconstruct the learned sequence(s). It is a fascinating direction for future research to devise a coercion-resistant system where an implicitly learned secret is used in a challenge-response protocol with the server. We come back to this question at the end of the paper.

5.2 An Experiment: Extracting Sequence Fragments

One of the potential attacks on our system involves a malicious party profiling the legitimate user’s knowledge and using that information to reverse engineer the trained sequence to be able to pass the authentication test. Although the number of possible trained sequences is too large to exhaustively test on any single individual each sequence is constructed according to known constraints and knowledge of subsequence fragments might enable the attacker to either reconstruct the original sequence or enough of it to pass an authentication test.

The training sequences are constrained to use all 6 response keys equally often, so analysis of individual response probabilities cannot provide information about the trained sequence. Likewise all 30 possible response key pairs ($6 * 5 = 30$, since keys are not repeated) occur equally often during training meaning that bigram frequency also provides no information about the trained sequence. However, each 30-item sequence has 30 unique trigrams (of 150 possible). If the specific training trigram fragments could be identified, the underlying training sequence could be reconstructed.

An attack based on this information would be to have a trained user perform a SISL test that contains all 150 trigrams equally often. If the user exhibited better performance on the 30 trained trigrams than the 120 untrained, the sequence could be reconstructed. This attack would weaken the method’s relative resistance to external pressure to reveal the authentication information.

However, while the sequence information can be determined at the trigram level it is not known if participants reliably exhibit sequence knowledge in such short fragments. In Experiment 3, we evaluated performance on this type of trigram test to assess whether the sequence information could be reconstructed.

Participants were again recruited through Mechanical Turk and completed the same training sessions used in Experiments 1 and 2. At test, participants performed a sequence constructed to provide each of the 150 trigrams exactly 10 times by constructing ten different 150-trial units that each contain all possible trigrams in varying order. Performance on each trigram was measured by percent correct as a function of the current response and two responses prior.

To evaluate whether these data could be used to reconstruct the sequence, the percent correct on each trigram was individually calculated and a rank order of all trigrams was created for each individual. If performance on the trained trigrams was superior to others, the trained trigram ranks should tend to be lower (e.g., performance expression would lead the sequence trigrams to be the 30 best performed responses). However, average rank and average percent correct on the trained trigrams was indistinguishable from untrained trigrams. Participants did not exhibit their trained sequence knowledge on this type of test, indicating that their sequence knowledge cannot be attacked with a trigram-based method. More specifically, for each user we compared the average percent correct measurements for the 30 trained-sequence trigrams to those for the 120 remaining trigrams. The 34 participants averaged 73.9% correct (SE 1.2%) for trigrams from the trained sequence and 73.2% correct (SE 1.1%) for the rest. The difference was not reliable.

While the trigram test did not lead to expression of sequence knowledge, it is likely that participants' sequence knowledge could be assessed for some longer fragments. However, the number of fragments to assess grows exponentially with the length to be assessed and the ability to test all fragments is limited by the need to rely on human performance to do the assessment. For example, for length 4 fragments (quad-grams), there are 750 fragments to assess multiple times each to try to identify which ones had been trained.

Future work. In future work we will assess sequence expression at various lengths to find the minimal length at which sequence knowledge can be expressed. This minimal length likely reflects a basic operating characteristic of the brain regions that support implicit sequential skill learning. If this length suggests the possibility of attack, the sequence can be increased in complexity by increasing the number of characters, using inter-response timing (known to be important to learning [7]) or more complex sequence structures than simple repeating sequences.

Recall that in our experiments we assumed that users are honest during the training phase and the adversary only gets to coerce users after they have been trained.

We leave it for future work to design a coercion-resistant authentication protocol that remains secure when users can be coerced during the training phase.

6 Related Work

There is a large body of related work in user authentication and biometrics for user access control. The work can be broken down into biometrics (“who you are”), tokens (“what you have”), and passwords (“what you know”). There is significant past work in each of the three main areas. Our work may fall into a new category of implicit learning (“what you know you know but do not know”), or could be categorized as a subclass of behavioral biometric measurement.

Classic biometrics identifying a user based on who they physically are can be grouped into physiological and behavior categories. Physiological characteristics include fingerprint, face recognition, DNA prints, palm print, hand geometry, iris recognition, and retinal scans. Behavioral characteristics include measurements of typing rhythm and other dynamics, dynamic signature, walking gait, voiceprints, and eye movement patterns [11, 10, 2, 15]. Our work differs from these in enabling quick training in new randomly seeded patterns. It might be very difficult to learn to walk a new way, and nearly impossible to change one's iris pattern, but it should be easy to learn a new cortical crypto sequence with a modest training regime. Further, if one relies on retinal patterns for identification, each system could capture all the information content of the retina, and thus a single compromised retina reader could reveal to an adversary the entire set of information. Our approach enables key revocation and multiple keys per user for different systems where there need not be any information leakage from one system to the next.

Denning et al. [4] propose an authentication model based on implicit learning of sets of images. An earlier study [21] compared the learning of images, artificial words, and outputs from finite-state automata. Both of these works develop authentication systems that allow users to easily memorize strong passwords, however the resulting systems are not as resistant to rubber hose attacks because they depend on the user consciously studying sets of images or strings and as a result the user retains some conscious knowledge of the credential. When using the SISL task we were able to verify that little conscious knowledge of the trained secret is retained. Image-based authentication mechanisms also require curated image sets in order to reduce errors in the authentication process; in contrast SISL-based authentication uses automatically generated sequences sampled from a well-defined high entropy combinatorial space.

Deniable encryption. In the context of encryption, *deniable encryption* [3, 13] enables a user who encrypts a message to open the ciphertext in multiple ways to produce different cleartexts from the same ciphertext. Such systems enable a user to reveal an encryption key, which produces a document that contains plausible cleartext, but which is different from the actual document the user wishes to protect. This technique protects encrypted documents, but does not apply to authentication credentials. Further, a properly motivated user of deniable encryption could choose to reveal the correct decryption key, enabling the coercive adversary offline access to all versions of the document. Our approach develops a system where the user cannot, even if strongly motivated, reveal to another any information useful for an adversary to replicate the user’s access without the user being present. Deniability has also been studied in the context of elections [9].

Coercion detection. Since our aim is to prevent users from effectively transmitting the ability to authenticate to others, there remains an attack where an adversary coerces a user to authenticate while they are under adversary control. It is possible to reduce the effectiveness of this technique if the system could detect if the user is under duress. Some behaviors such as timed responses to stimuli may detectably change when the user is under duress. Alternately, we might imagine other modes of detection of duress, including video monitoring, voice stress detection, and skin conductance monitoring [8, 16, 1]. The idea here would be to detect by out-of-band techniques the effects of coercion. Together with in-band detection of altered performance, we may be able to reliably detect coerced users.

7 Conclusions and Future Work

We have presented a new approach to protecting against coercion attacks using the concept of *implicit learning* from cognitive psychology. We described a proof of concept protocol and preliminary experiments conducted through Mechanical Turk demonstrating a basis for confidence that it is possible to construct rubber hose resistant authentication.

Much work remains. We hope to further analyze the rate at which implicitly learned passwords are forgotten, and the required frequency of refresher sessions. In addition we would like to find methods to detect or predict when individual users reliably learn (collecting more demographic data about our users might be a good first step in this direction, along with multi-session long-term experiments). We also hope to explore some of the limits of the approach, for example by finding out the minimum lengths at which parts of learned sequences are distin-

guishable to an attacker versus a legitimate authenticator, as well as by strengthening the test procedures and analysis to increase reliability across a larger fraction of users, or reduce the required testing time, false positives, and false negatives. Using variable timing between cues and measuring user performance as a function of game speed can further help in making the test protocol more reliable. Implicit learning of multiple credentials is yet another area that can benefit from additional experiments, building upon prior work that has so far found no evidence of interference when users learn distinct 12-item sequences, while also being capable of learning implicitly sequences as long as 80 items.

Another future direction for this work is in testing whether more complex structures—for example Markov models—can be learned implicitly. We would like to use such learning to build challenge-response authentication which is resistant to eavesdropping in addition to coercion. Finally, beyond authentication, we would like to investigate the construction of a variety of cryptographic primitives based on implicit learning.

Acknowledgment

We would like to thank all the paid volunteers who have contributed to our user studies through their participation. This work was funded by NSF and a MURI grant.

References

- [1] J. Benaloh and D. Tuinstra. Uncoercible communication. Technical Report TR-MCS-94-1, Clarkson University, 1994.
- [2] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 568–574, 1997.
- [3] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *CRYPTO*, pages 90–104, 1997.
- [4] Tamara Denning, Kevin D. Bowers, Marten van Dijk, and Ari Juels. Exploring implicit memory for painless password recovery. In Desney S. Tan, Saleema Amershi, Bo Begole, Wendy A. Kellogg, and Manas Tungare, editors, *CHI*, pages 2615–2618. ACM, 2011.
- [5] A. Destrebecqz and A. Cleeremans. Can sequence learning be implicit? new evidence with the process dissociation procedure. *Psychonomic Bulletin & Review*, 8:343–350, 2001.

- [6] Julien Doyon, Pierre Bellec, Rhonda Amsel, Virginia Penhune, Oury Monchi, Julie Carrier, Stéphane Lehericy, and Habib Benali. Contributions of the basal ganglia and functionally related brain structures to motor learning. *Behavioural Brain Research*, 199(1):61–75, April 2009.
- [7] E. Gobel, D. Sanchez, and P. Reber. Integration of temporal and ordinal information during serial interception sequence learning. *Journal of Experimental Psychology: Learning, Memory & Cognition*, 37:994–1000, 2011.
- [8] Payas Gupta and Debin Gao. Fighting coercion attacks in key generation using skin conductance. In *USENIX Security Symposium*, pages 469–484, 2010.
- [9] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 61–70, New York, NY, USA, 2005. ACM.
- [10] A. Kale, A.N. Rajagopalan, N. Cuntoor, V. Krueger, and R. Chellappa. Identification of humans using gait. *IEEE Transactions on Image Processing*, 13:1163–1173, 2002.
- [11] Fabian Monrose, Michael Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *Int. J. of Inf. Sec.*, 1(2):69–83, 2002.
- [12] Mary J. Nissen and Peter Bullemer. Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, 19(1):1–32, January 1987.
- [13] Adam O’Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. In *Proc. of Crypto’11*, volume 6841 of *LNCS*, pages 525–542, 2011.
- [14] Paul Reber. Cognitive neuroscience of declarative and non-declarative memory. *Parallels in Learning and Memory*, Eds. M.Guadagnoli, M.S. deBelle, B. Emyre, T. Polk, A. Benjamin, pages 113–123, 2008.
- [15] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing*, 2000.
- [16] Robert Ruiz, Claude Legros, and Antonio Guell. Voice analysis to predict the psychological or physical state of a speaker, 1990.
- [17] D. Sanchez, E. Gobel, and P. Reber. Performing the unexplainable: Implicit task performance reveals individually reliable sequence learning without explicit knowledge. *Psychonomic Bulletin & Review*, 17:790–796, 2010.
- [18] D.J. Sanchez and P.J. Reber. Operating characteristics of the implicit learning system during serial interception sequence learning. *Journal of Experimental Psychology: Human Perception and Performance*, in press.
- [19] Chris Soghoian. Turkish police may have beaten encryption key out of TJ Maxx suspect, 2008. news.cnet.com/8301-13739_3-10069776-46.html.
- [20] T. van Aardenne-Ehrenfest and N. G. de Bruijn. Circuits and trees in oriented linear graphs. *Simon Stevin*, 28:203–217, 1951.
- [21] Daphna Weinsshall and Scott Kirkpatrick. Passwords you’ll never forget, but can’t recall. In *CHI Extended Abstracts*, pages 1399–1402, 2004.
- [22] Wikipedia. Rubber-hose cryptanalysis, 2011.