October 2019

Authentication Usability Methodology

Jean-Baptiste Subils
University of South Florida

Authentication Usability Methodology

by

Jean-Baptiste Subils

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Jay Ligatti, Ph.D.
Dmitry Goldgof, Ph.D.
Ou Xinming, Ph.D.
Sean Barbeau, Ph.D.
Kaiqi Xiong, Ph.D.

Date of Approval:
October 22, 2019

Keywords: Security, Access Control, Dual-Task, Gamification, Cognitive Load

Table of Contents

List of Tables

List of Figures

Abstract

Nowadays many systems require end users to authenticate themselves. Authentication is one of the security activities that end users perform the most. Thus, the usability of this security feature plays a major role in the proper utilization and adoption of a novel authentication method.

This dissertation presents coauthentication, a novel authentication system. Many authentication methods and protocols exist, but passwords remain the predominant authentication method used. Coauthentication is presented here in detail in several possible variations and their associated protocols, with performance comparisons.

This dissertation also presents a framework to evaluate authentication methods in terms of usability. A large body of literature pertaining to the usability of computer systems is available; however, comparing the usability of authentication methods remains difficult due to the different techniques available. Several usability methodologies are reviewed as well as several overall comparison tools used to compare authentication methods.

A study of 43 participants, following the framework presented, evaluates coauthentication against passwords on two different entry devices, a laptop and a smartphone, and against fingerprints on a smartphone. The study results provide a promising framework for comparing usability of authentication techniques.

Chapter 1: Introduction

Authentication is one of the most common security activities end users perform. Due to this activity being reoccurring, time consuming, and often considered annoying [2], users tend to prefer authentication with minimal effort on their part. Thus, in order for end users to adopt a new authentication method, usability is crucial. Due to the usability and popularity of passwords, a novel authentication method also needs to yield significantly better results than a password-based authentication method [3].

1.1 Coauthentication's Background

As is well understood, user authentication is based on factors, the three standard factors being what you know (human-entered secrets like passwords), what you have (physical tokens like keys, electronic remote controls, or smartcards), and what you are (biometrics like fingerprints). Every authentication system, regardless of the factors used, is based on secrets, which could take the form of passwords, patterns of metallic teeth on keys, radio frequencies at which devices transmit data, codes stored on devices and transmitted, fingerprints, etc. Authentication systems aim to protect against attackers who have not obtained the required secrets.

Each authentication factor has advantages and disadvantages [4]. For example, tokens are susceptible to theft, but doing so in the obvious way requires physical access. Users will often notice physical theft of a token more readily than a remote theft or guessing of a password or biometrics. However, tokens have traditionally relied on special-purpose hardware and consequently been more expensive to implement and deploy than other factors. In addition, usability benefits of tokens have traditionally been offset by the costs of having to carry and handle the tokens [4, 5]. Some attacks are also capable of bypassing authentication requiring a password [6, 7].

Multi-factor authentication attempts to improve security by requiring successful attacks to compromise every factor being used [8]. One two-factor mechanism combines a username and password with a second password (a one-time password, OTP) texted to the user's phone [9]. Alternatively, instead of receiving an OTP from the authenticator, the phone may share a cryptographic key with the authenticator and generate its own OTP, called a time-based OTP or TOTP, as a cryptographic hash, using the shared key, of the current time [10]. A benefit of such mechanisms is that the physical-token factor is a device already possessed and carried by the user, thus avoiding expensive, dedicated hardware.

However, multi-factor techniques add the inconveniences of each factor required. For example, because OTP and TOTP techniques require users to enter two passwords and carry a registered device, they suffer from the nontrivial usability drawbacks of password-based authentication mechanisms (e.g., [11, 12, 13, 14, 15]) and the inconvenience of having to access a mobile device to authenticate.

This latter inconvenience, of having to access one's registered mobile device to authenticate, has lessened over time, as the overwhelming majority of adults have gone from having zero personal smart devices accessible at all times to having one personal smart device—a smartphone—accessible at all times [16].

With the growth of the Internet of Things, ubiquitous computing, and wearable, edible, and implantable devices, the overwhelming majority of adults may soon have multiple personal smart devices accessible at all times, all of which can be registered and used to authenticate. For example, to log in to a website, open a door, or start an engine, *two* of a user's registered devices, perhaps a smartphone and smartwatch, might participate in the authentication. A gate or garage door might authenticate a request to open by requiring participation from both a registered car and a registered smartphone; then stealing only the car, or only the phone, would be insufficient for opening the door.

Even today many people only authenticate to certain services when multiple of their devices are present. For example, a user U may log in to banking services only from a certain PC while in the presence of U's smartphone. In this case the banking service could register these two user devices to U and require their participation in every authentication of U. Because the PC and smartphone are separate and heterogeneous, successfully stealing or otherwise attacking one device does not imply a successful attack on the other device. It is therefore of value to protect against attacks on only one of the two user devices.

We call this single-factor technique, in which multiple devices collaborate to authenticate a user, *coauthentication*. The user devices collaborate through cryptographic protocols, such that an authenticator receives message(s) proving that all required user devices approve the authentication. Attackers who steal only one of the user devices cannot authenticate, because the unstolen device will not approve the authentication.

Benefits of coauthentication include protecting against the compromise of authentication secrets (cryptographic keys); preventing phishing, replay, and man-in-the-middle attacks; basing authentication on high-entropy secrets that can be generated and updated automatically; avoiding the inconveniences of factors like passwords and biometrics; implementing advanced authentication functionalities, including m-out-of-n, continuous, group, shared-device, and anonymous authentication; and, when implementing m-out-of-n authentication, providing availability protections against device misplacement and denial-of-service attacks.

1.2 Authentication Usability Methodology

As explained previously authentication methods are based on three standard factors. Due to the fundamental differences between these factors, comparing authentication methods using different factor(s), especially in terms of usability, is not trivial.

Bonneau et al. introduced a subjective framework which provides a qualitative scale to compare authentication methods [3]. While offering a good assessment of authentication methods under a thorough list of categories, this framework remains a high-level overview for each category (e.g., usability). Usability is highly subjective, and thus, depends on the perception of users. Therefore, feedback from participants is necessary to obtain a more precise comparison. Another standard approach to compare authentication methods is the System Usability Scale (SUS), which is an effective metric for usability comparison [17, 18]. However, SUS only captures the satisfaction aspect of usability. SUS collects feedback from users through a questionnaire answered via a Likert scale that provides a score from $0 - 100$ [19].

The experimental framework presented provides a solution to compare all authentication methods in terms of usability, regardless of the authentication factor(s). The framework incorporates the System Usability Scale and other metrics (e.g., time of completion, accuracy) to compare authentication methods.

1.3 The Framework's Motivation and Background

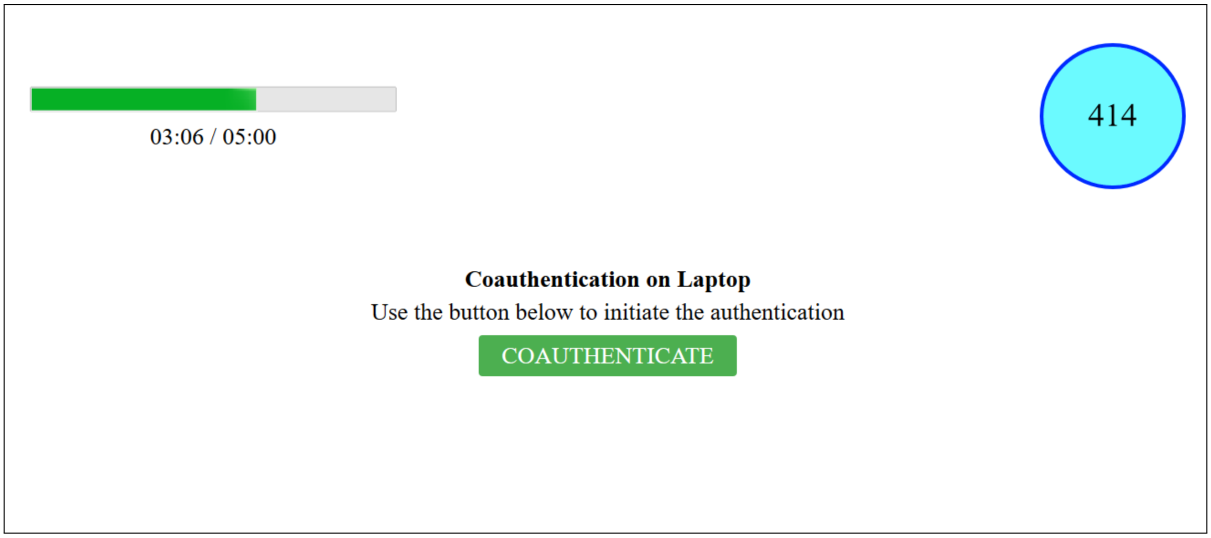
This section introduces the design principles of the framework. The framework presented in here focuses on the usability aspect of authentication methods and incorporates multiple metrics that can be applied for usability comparison [20].

To collect ecologically sound data, a scenario close to reality needs to be designed. Users do not choose to authenticate but are, instead, interrupted by authentication requirement(s). Authentication happens when users are accessing some resources requiring them to prove their identity. Thus, an activity representing a specific action being interfered with should be defined to reproduce a real use of an authentication method. Participants can attempt to complete the activity while being obstructed by authentication requirements. For example, users may be required to authenticate when accessing a website, while calling a support service, participating in a teleconference, or carrying on a conversation.

Authentication usually interferes with the access of some resources, thus, dual-task interference is a suitable technique to study user perception of authentication methods [21, 22]. In the field of cognitive psychology, dual-task interference is used to determine a person's cognitive load and ability to multitask. Due to the results of dual-task interference, the framework requires participants to undergo a dual-task interference game in addition to performing each authentication technique in a standalone manner.

The framework should collect information on authentication methods performed in a standalone manner to provide a baseline to compare with the data collected during the dual-task interference game (i.e., DTI game). This part serves as a training phase and allows participants to get accustomed to each of the authentication techniques evaluated. Prior to the DTI game, another training phase is helpful to familiarize participants with the game. The data resulting from this training can also be helpful in detecting improvements. The training phases should require participants to complete, first the activity, second the authentication tasks, and finally, both simultaneously to engage with the DTI game.

Gamification is a compelling incentive that pushes participants to complete experiments, and, in turn, produces meaningful data in case studies [23, 24, 25]. The experiment includes a multi-tasking game to actively engage participants. A scoreboard is a type of gamification element that gives feedback to the participant on their performance and provides an incentive to surpass themselves. The participant's score should be updated in real time to provide direct feedback. Participants' compensation can be based on their score, as an additional compelling incentive. Figure 1.1 shows a screenshot of the web application used, with the progress bar on the top left, the participant's score in the circle on the top right, and in the middle the authentication requirement.



Coauthentication on Laptop

Use the button below to initiate the authentication

Figure 1.1: Screenshot of the multitasking game while authenticating via coauthentication.

1.4 Contributions

1.4.1 Coauthentication's Contributions

Coauthentication is the first single-factor, multi-device technique for authenticating users without passwords or biometrics.

Chapter 2 introduces and evaluates coauthentication, including several specific coauthentication system designs, protocols, and implementations. It makes the following contributions.

- Example coauthentication system designs, attack models, policies, and applications are presented (Section 2.1).

- Coauthentication protocols, having strong two-way authentication and forward-secrecy properties, are defined (Section 2.2).

- The principal security properties of the coauthentication protocols are formally verified, using ProVerif [26, 27], under a small set of explicitly stated, realistic assumptions (Section 2.3).

- The implementability and performance of the coauthentication protocols are evaluated (Section 2.4).

- Several extensions and generalizations of coauthentication are provided (Section 2.5).

- In a discussion of related work, it is shown that existing authentication techniques, specifically those like OTPs that may involve multiple user devices, can also benefit from the coauthentication protocols (Section 2.6).

Section 5.2 concludes and Section 5.3.1 describes future work.

1.4.2 The Framework's Contributions

The framework is used to evaluate the following usability aspects of authentication methods:

- Efficiency, which is the authentication's completion time.

- Effectiveness, which is the success rate.

- Satisfaction, which is rated via participants' feedback.

Efficiency is defined as the length of time necessary for a user to be authenticated. Thus, this time is calculated from the first user action performed to authenticate until the user receives the authentication result. Additionally, this time takes into account the user-interaction time required to perform an authentication task.

To compare authentication methods fairly, regardless of the authentication factor and in terms of usability, a success rate is essential. False positives directly affect the usability of an authentication method, and in practice, re-authenticating due to a failed attempt, with a certain limit to prevent brute-force attacks, is allowed to improve usability. The purpose for allowing retries is to reduce the inconvenience induced from forcing users to re-authenticate. False negative and false positive rates are metrics used to determine the security and usability of biometric authentication methods [28]. However, other types of authentication factors are less subject to accuracy problems. Accuracy will be used as a metric to determine the effectiveness of each authentication method evaluated.

Satisfaction is the subjective perception of usability. To collect participants' feedback the framework uses the Authentication Experience Questionnaire (see Appendix B), which includes the System Usability Scale (SUS) questionnaire and various feedback questions. SUS is a widely used and accepted approach to determine the usability of a computer system [17, 29].

Chapter 2: Coauthentication

Coauthentication is the first single-factor, multi-device technique for authenticating users without passwords or biometrics [30, 31, 32, 33, 34].

To provide the reader with a more complete understanding of coauthentication and its benefits this dissertation also reports on the results of work performed by Cagri Cetin [35]. Specifically Cagri Cetin implemented the formal verification and performed the evaluation as part of the coauthentication project, in which we both participated.

This section introduces and evaluates coauthentication, including several specific coauthentication system designs, protocols, and implementations. It makes the following contributions.

- Example coauthentication system designs, attack models, policies, and applications are presented (Section 2.1).

- Coauthentication protocols, having strong two-way authentication and forward-secrecy properties, are defined (Section 2.2).

- The principal security properties of the coauthentication protocols are formally verified, using ProVerif [26, 27], under a small set of explicitly stated, realistic assumptions (Section 2.3).

- The implementability and performance of the coauthentication protocols are evaluated (Section 2.4).

- Several extensions and generalizations of coauthentication are provided (Section 2.5).

- In a discussion of related work, it is shown that existing authentication techniques, specifically those like OTPs that may involve multiple user devices, can also benefit from the coauthentication protocols (Section 2.6).

2.1 Coauthentication System Designs, Policies, and Applications

The devices involved in coauthentication are the *authenticator* (e.g., a server deciding whether to authenticate a user), the *requestor* (on which the current authentication attempt is initiated), and one or more *collaborators*. The requestor and collaborator(s) are *registered* with the authenticator, meaning that the devices have access to a secret that the authenticator can use to verify the devices' participation in an authentication. This secret accessible to the requestor and collaborator(s) may, for example, be a secret key shared with the authenticator, or a private key K such that the authenticator can verify signatures created with K.

In some coauthentication protocols, the authenticator, upon receiving an authentication *request*, issues one or more *challenges* and awaits one or more valid *responses* to the challenges. Other protocols avoid authentication challenges. In all cases, the authenticator verifies that multiple registered devices, more specifically the secret keys accessible to those devices, participate in the authentication.

2.1.1 Attack Models and Assumptions

Coauthentication, like multi-factor techniques, protects against theft of any one authentication secret. The secrets in coauthentication are cryptographic keys. Theft of coauthentication secrets may occur in any way, including by remotely compromising devices to obtain their stored keys or physically stealing devices.

Attackers are assumed to be active and can eavesdrop on, insert, delete, and modify communications. Attackers may mount replay and man-in-the-middle attacks.

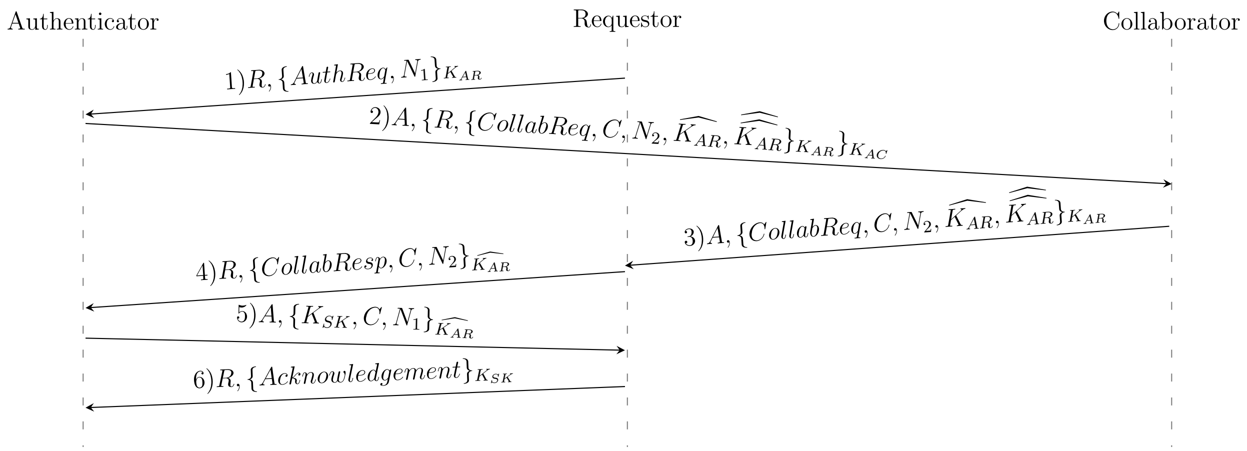

Authenticator
Requestor
Collaborator
1)$R, \{AuthReq, N_1\}_{K_{AR}}$
2)$A, \{R, \{CollabReq, C, N_2, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}\}_{K_{AC}}$
3)$A, \{CollabReq, C, N_2, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}$
4)$R, \{CollabResp, C, N_2\}_{\widehat{K_{AR}}}$
5)$A, \{K_{SK}, C, N_1\}_{\widehat{K_{AR}}}$
6)$R, \{Acknowledgement\}_{K_{SK}}$

Figure 2.1: The full coauthentication protocol. Secret key K_{AR} (K_{AC}) is shared between authenticator and requestor (collaborator). Each N_i is a nonce, and $\{M\}_K$ is the encryption of M using key K. The third message is sent through a private channel.

Attackers are however assumed to be incapable of cryptanalysis; attackers can only infer plaintexts from ciphertexts when also having the required secret key. Without such an assumption, attackers could extract credentials like session keys simply by monitoring and cryptanalyzing legitimate authentications.

Some coauthentication protocols protect against attackers who know all the secrets stored on a device that the victim user possesses. We call such attacks *key-duplication attacks*. For example, an attacker may duplicate a device's secret keys by remotely compromising the device. Alternatively, the attacker may physically steal a device, duplicate all keys accessible to the device, and return the device to the victim user, who may be unaware of the theft and duplication.

To protect against key-duplication attacks, the coauthentication protocols assume that a private communication channel, inaccessible to attackers, exists between the requestor and collaborator devices. Such an assumption is necessary because the duplicated keys must be updated through some channel inaccessible to the attacker; otherwise, the attacker—who has all of the victim device D's keys—could decrypt and obtain any updated keys sent to D, and modify any updated keys sent from D. Private channels may be implemented with short-range communications, such as NFC, zigbee, wireless USB, infrared, or near-field magnetic induction, under the assumption that attackers cannot access such communications because they are on direct, device-to-device channels.

Other coauthentication protocols do not require a private channel between requestor and collaborator devices. Although these protocols do not protect against key-duplication attacks, they do protect against attackers who obtain keys by stealing devices (without duplicating the keys in, and returning, the devices). In other words, the attack model for these all-public-channel protocols assumes that if an attacker has obtained a device D's authentication secret, then D's legitimate user no longer possesses D.

All of this dissertation's coauthentication protocols assume that devices in the user's possession run as intended during the coauthentication process. Without such an assumption, malware on the user's requestor device could simply leak decrypted session keys or any other unencrypted private data, and malware on the user's collaborator device could simply approve an attacker's authentication requests. Protecting against malware that is actively running on a device in the user's possession, while the user is authenticating, is beyond the scope of coauthentication.

All of this dissertation's coauthentication protocols also assume that authenticators run as intended during the coauthentication process. Without such an assumption, malware on the authenticator could simply leak secrets or allow all authentication requests. Protecting against malware on authenticators is beyond the scope of coauthentication.

2.1.2 Collaboration Policies

Each collaborator may enforce its own policy defining the circumstances under which it participates in a coauthentication.

For example, a collaborator may only participate in an authentication after a user has clicked a button or provided some other input to confirm participation. Under this policy, if an attacker steals or compromises the requestor and initiates a coauthentication, the legitimate user will not confirm the attacker-initiated authentication on the collaborator, so the authentication attempt will fail.

Alternatively, a collaborator may automatically participate in an authentication but warn the user, or log, that it has done so, for example by displaying a text alert with an audible warning sound (e.g., a text message). The alert could provide a simple interface for the user to notify the authenticator if the collaboration was unauthorized (i.e., an attacker-initiated authentication).

The first of these example policies, which we call the *disallow-by-default* collaboration policy, only collaborates when a user confirms the authentication. The second policy, which we call the *allow-by-default-with-warning* collaboration policy, relies on users to observe a warning and handle unauthorized collaborations after the fact. For many applications the usability benefits of the allow-by-default-with-warning policy may outweigh the security costs; many modern authentication systems email or text users after suspicious logins and request after-the-fact notification of unauthorized access.

Additional collaboration policies are possible. For example, a collaborator could decide whether to participate in a coauthentication based on the requestor's proximity, that is, whether the requesting device is co-located with the collaborator. In applications where the attack vector of concern is device theft, a collaborator may presume that a co-located requestor has not been stolen. Such a collaborator may tacitly allow collaborations with co-located requestors but show warnings for, require explicit confirmations for, or disallow entirely, collaborations with non-co-located requestors.

When run automatically, without requiring user interaction, coauthentication is a zero-interaction authentication system [36]. Zero-interaction systems are well suited to continuous authentication [36, 37].

2.2 The Full Coauthentication Protocol

Figure 2.1 illustrates the full coauthentication protocol for two user devices. Authentication secrets in this protocol are shared symmetric-cryptography keys, and there is only one collaborator.

Following the flow of data in Figure 2.1, the full protocol operates as follows. Assume that during device registration, the authenticator A and requestor R share a secret key K_{AR}, and the authenticator A and collaborator C share a secret key K_{AC}.

1. Requestor R initiates the coauthentication by sending the authenticator A its ID and an encrypted authentication-request message containing a challenge nonce N_1 (which serves to authenticate A to R).

2. Authenticator A receives and decrypts the request message, finds that the requestor R is registered to a user having collaborating device C, creates a challenge nonce N_2 (which serves to authenticate R to A), generates *two new keys* ($\widehat{K_{AR}}$ and $\widehat{\widehat{K_{AR}}}$) to share with R (to rotate keys, to ensure forward secrecy and prevent key-duplication attacks), and double encrypts these data in a collaboration-request message to C, the first (inner) encryption using K_{AR} and the second (outer) encryption using K_{AC}. By double encrypting nonce N_2, the authenticator ensures participation of both user devices' secret keys (K_{AR} and K_{AC}) in the coauthentication.

3. Collaborator C receives and decrypts the previous message, verifies the identity of the requestor, and forwards the decrypted message (which is still ciphertext encrypted with K_{AR}) to requestor R through a private channel.

4. Requestor R receives and decrypts this message using K_{AR}, verifies the identity of the collaborator, and obtains N_2, $\widehat{K_{AR}}$, and $\widehat{\widehat{K_{AR}}}$. The requestor then generates and sends the authenticator a collaboration-response message containing N_2 encrypted with its first updated key, $\widehat{K_{AR}}$. The requestor saves the second updated key, $\widehat{\widehat{K_{AR}}}$, for a future coauthentication request.
5. Authenticator A receives the collaboration-response message, decrypts, and verifies the collaborator's identity and that the received nonce matches the N_2 it sent earlier. Because A has now verified participation of both keys K_{AR} and K_{AC}, it sends an authentication-complete message, for example containing a session key K_{SK}, to the requestor R.
6. Requestor R sends an acknowledgment to the authenticator.

Timestamps may be added to these messages, for example to implement timeouts or fine-grained logging.

Notice that full coauthentication stores three keys long term: K_{AR} may be stored long term before the current round of authentication, $\widehat{\widehat{K_{AR}}}$ may be stored long term after the current round of authentication, and K_{AC} may be stored long term before and after the current round of authentication.

2.2.1 Properties of the Full Protocol

The full coauthentication protocol uses nonces to authenticate the requestor and authenticator to each other—session keys are only shared between mutually authenticated devices. Requestor R only shares session keys with authenticated As, and authenticator A only shares session keys with authenticated Rs.

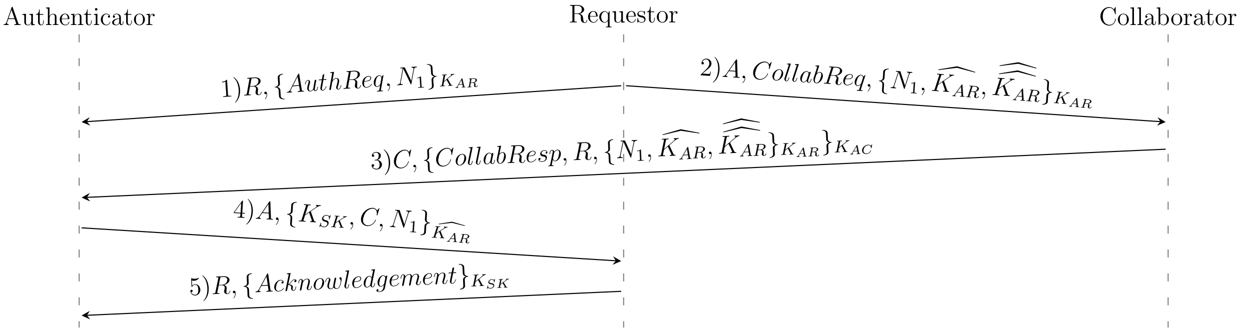

Authenticator
Requestor
Collaborator
1)$R, \{AuthReq, N_1\}_{K_{AR}}$
2)$A, CollabReq, \{N_1, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}$
3)$C, \{CollabResp, R, \{N_1, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}\}_{K_{AC}}$
4)$A, \{K_{SK}, C, N_1\}_{\widehat{K_{AR}}}$
5)$R, \{Acknowledgement\}_{K_{SK}}$

Figure 2.2: A coauthentication protocol omitting authenticator challenges. The second message is sent through a private channel.

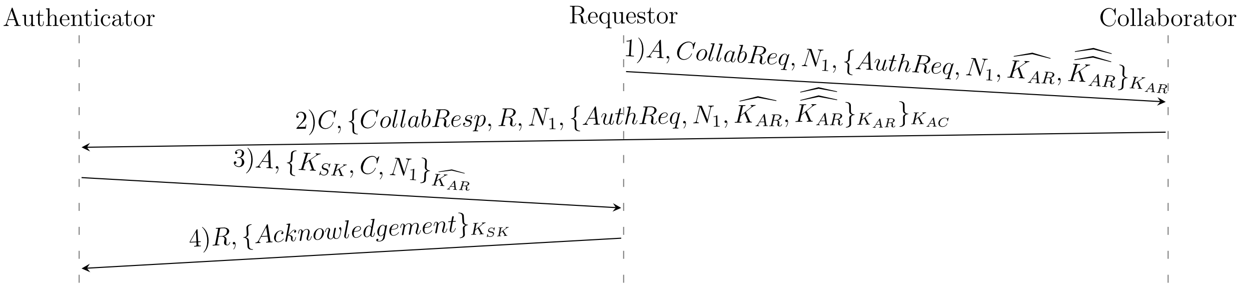

Authenticator
Requestor
Collaborator
1)$A, CollabReq, N_1, \{AuthReq, N_1, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}$
2)$C, \{CollabResp, R, N_1, \{AuthReq, N_1, \widehat{K_{AR}}, \widehat{\widehat{K_{AR}}}\}_{K_{AR}}\}_{K_{AC}}$
3)$A, \{K_{SK}, C, N_1\}_{\widehat{K_{AR}}}$
4)$R, \{Acknowledgement\}_{K_{SK}}$

Figure 2.3: A challengeless coauthentication protocol incorporating message forwarding. The first message is sent through a private channel.

The full protocol also employs key rotation to ensure forward secrecy. An attacker who acquires the keys stored long term on at most one user device cannot obtain past session keys. Each session key K_{SK} is encrypted with an updated $\widehat{K_{AR}}$.

The full protocol mitigates man-in-the-middle attacks by making the authentication secrets shared between the authenticator and user devices be cryptographic keys, used to encrypt communications. In contrast, man-in-the-middle attacks may be possible on password or biometrics systems because the authenticator may only share, with users or user devices, secrets that are insufficient for cryptographic use. For example, a man-in-the-middle attack on an OTP system may proceed as follows: the victim enters a username and password on a fake website; the fake website forwards this information to the real website, which then issues an OTP; the victim receives and enters the OTP into the fake website; the attacker completes the authentication on the real website and masquerades as the user. In this case the shared username/password (or hash thereof) is insufficient for providing the cryptographic properties needed to mitigate man-in-the-middle attacks.

Now suppose an attacker acquires the long-term secrets stored on at most one user device. Acquiring K_{AC} only enables an attacker, even one with access to the private channel, to permit or deny authentications initiated by the victim. Attackers are already assumed to be active and consequently capable of denying service by dropping network messages. Acquiring K_{AC} therefore provides an attacker with no new capabilities (and Section 2.5.1 describes extensions of coauthentication that mitigate denial-of-service attacks on user devices).

On the other hand, acquiring only the K_{AR} to be used in the next coauthentication request enables an attacker to request authentication, but assuming an appropriate collaboration policy, the collaborator will notify the victim user of the authentication attempt. From the victim's perspective, this attacker-initiated authentication attempt will be unexpected, so the victim will deny collaboration and therefore the authentication.

Acquiring only the K_{AR} to be used in the next coauthentication request also enables an attacker mounting a key-duplication attack to wait for and decrypt a legitimate authentication request coming from the requestor device, still in the victim's possession. However, such an attacker only obtains nonce N_1 in the process and cannot decrypt any of the remaining messages in the protocol, because they are either encrypted with different keys or sent on a private channel. Obtaining K_{AR} and N_1 provides an attacker with no new capabilities.

The full coauthentication protocol therefore protects against attackers who have acquired the long-term secrets stored on at most one user device. ProVerif has been used to formalize and verify these arguments, as described in Section 2.4.

2.2.2 Variation: Omitting the Challenge-Response

It is possible to avoid the challenge-response portion of the full coauthentication protocol, implemented with nonce N_2, by having the requestor send two requests, one to the authenticator (to request authentication) and another to the collaborator (to request collaboration).

Figure 2.2 shows such a challengeless protocol. The requestor sends two requests, one to the authenticator and another to the collaborator, containing the same nonce N_1. The requestor also includes the updated versions of K_{AR} in its collaboration-request message, which the collaborator forwards to the authenticator. These updated keys are double encrypted during transit from the collaborator to the authenticator, protecting the keys against attackers having obtained at most one of K_{AR} and K_{AC}. After verifying that both the requestor and its registered collaborator have participated in an authentication by sending the same N_1, the authenticator sends a new session key to the requestor, encrypted with the proper updated version of K_{AR}. As in the full protocol, this challengeless version results in the authenticator and requestor sharing an updated $\widehat{\widehat{K_{AR}}}$, usable in a subsequent run of the protocol as the new version of K_{AR}.

Having formally verified both the full and challengeless coauthentication protocols, to our knowledge they provide the same security guarantees. The known tradeoffs between these protocols relate to performance. The challengeless protocol is expected to be more efficient overall, due to the omission of challenge creation and the parallelization or batching of some of the communications (e.g., the first and second messages in Figure 2.2). However, the computations performed by individual devices may be more efficient in the full version. For example, from the requestor's perspective, the challengeless protocol essentially replaces the computations needed to decrypt the third message and generate the fourth message of Figure 2.1 with the computations needed to generate the second message of Figure 2.2, including generating updated versions of K_{AR}. For some user devices, such as IoT devices with limited resources, some of these computations may be more expensive than others, making one protocol more efficient than another for those devices.

2.2.3 Variation: Incorporating Message Forwarding

Figure 2.3 shows a variation of the challengeless protocol that incorporates message forwarding. The protocol shown in Figure 2.3 is the same as the one shown in Figure 2.2 but with the collaborator forwarding the authentication-request message to the authenticator on behalf of the requestor.

2.2.4 Variation: No Private Channels

In cases where a private channel does not exist between the requestor and collaborator, coauthentication protocols cannot prevent key-duplication attacks. The ability of an attacker, who has acquired all the secrets stored on a user-possessed requestor R, to eavesdrop on and modify all communications to and from R, makes it impossible to update R's secrets without the attacker also obtaining any updates sent to R and modifying any updates sent from R.

In practice it may be acceptable to dismiss key-duplication attacks by relying on alternative mechanisms to mitigate them. For example, a device's long-term, rarely updated key K_{AR} may be stored in a trusted platform module (TPM) [38]. With K_{AR} in a TPM, we might assume that attackers, who possibly have physical access to the requestor R, may be able to *use* K_{AR} to initiate authentications on R, but cannot *extract* K_{AR} from R. That is, mechanisms like TPMs may mitigate key-duplication attacks by allowing authentication secrets to be used but not extracted, and therefore not duplicated.

It may also be acceptable to dismiss key-duplication attacks in cases where the threat is considered remote or private channels simply cannot be implemented or would be costly to implement.

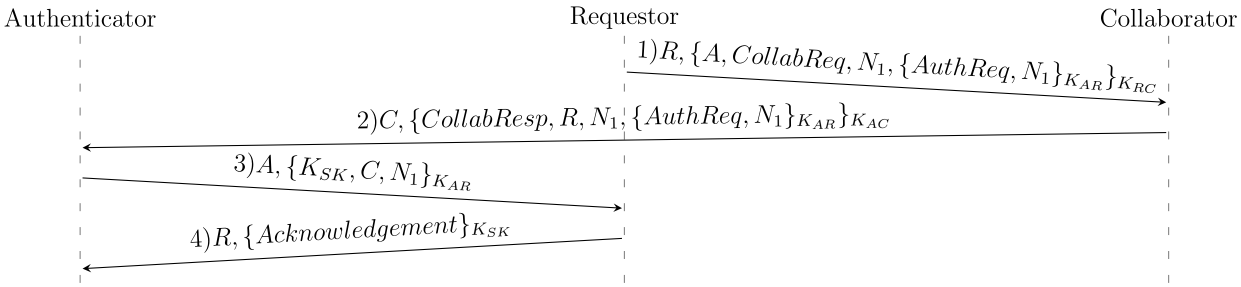

Authenticator
Requestor
Collaborator
$1)R, \{A, CollabReq, N_1, \{AuthReq, N_1\}_{K_{AR}}\}_{K_{RC}}$
$2)C, \{CollabResp, R, N_1, \{AuthReq, N_1\}_{K_{AR}}\}_{K_{AC}}$
$3)A, \{K_{SK}, C, N_1\}_{K_{AR}}$
$4)R, \{Acknowledgement\}_{K_{SK}}$

Figure 2.4: An all-public-channel variation of the challengeless coauthentication protocol with message forwarding (Figure 2.3).

In any of these cases, the coauthentication protocols can be varied to no longer require a private channel between the requestor and collaborator, yet still protect against non-key-duplication attacks. The attack model for these all-public-channel protocols assumes that if an attacker has obtained a device D's authentication secret, then D's legitimate user no longer possesses D. This attack model still covers attacks based on stealing devices and attempting to authenticate on the stolen devices.

For example, Figure 2.4 shows an all-public-channel variation of the protocol shown in Figure 2.3. The Figure-2.4 protocol matches the Figure-2.3 protocol, except that data for updating keys is omitted (because requestor-key updates cannot be confidential in an all-public-channel scenario in which an attacker has all of the requestor's secrets), and the message sent between the requestor and collaborator is encrypted with a shared key K_{RC} (because this message is sent over a public channel).

The all-public-channel protocols are simpler, and expected to run more efficiently, than the private-channel protocols but do not protect against key-duplication attacks and do not satisfy forward secrecy.

In practice a hybrid approach may be preferred: coauthentication keys may be updated only periodically, using private channels at opportune times, while public-channel protocols are used in the common case.

To make an analogy with password-based authentication systems, ideally—from a security perspective—users would update their passwords on every authentication, to limit attackers who have acquired passwords. Doing so would be like using the private-channel protocols for coauthentication. In practice, however, tradeoffs are made, and passwords are typically updated only rarely [12].

2.3 Formal Evaluation

The principal security properties of the example coauthentication protocols shown in Figures 2.1–2.4 have been formally verified with ProVerif [26, 27]. ProVerif uses a resolution-based strategy to verify that protocols satisfy desired security properties. A benefit of using ProVerif is that it can model arbitrarily many sessions of a protocol running concurrently.

Our ProVerif encodings of the coauthentication protocols, and the properties verified, are available online [39]. The protocol encodings faithfully follow the communications shown in Figures 2.1–2.4. The modeling of key updates uses key tables to store dynamically generated keys [40, p.37].

2.3.1 Assumptions

The protocols were modeled and verified under the assumptions stated in Section 2.1.1. The private-channel protocols (Figures 2.1–2.3) have strong attack models allowing key-duplication attacks.

The all-public-channel protocol (Figure 2.4) has a weaker attack model that assumes authentication secrets (K_{AR}, K_{AC}, and K_{RC}) are only accessible to attackers through device theft. In terms of the ProVerif encodings, this weaker attack model means that, in cases where attackers are assumed to know K_{AR}, the collaborator does not respond to collaboration requests. The justification is that if an attacker has acquired K_{AR}, then by assumption the legitimate user does not possess the requestor, so collaboration requests must be for unauthorized, attacker-initiated authentications. It is assumed that, with appropriate collaboration policies, users do not approve collaborations for unauthorized authentications.

In all the protocols, attackers are active and may freely eavesdrop on, insert, delete, and modify communications. Attackers are not constrained to operate according to any of the protocols.

In addition to arbitrary active attackers, each protocol session runs 3 processes (authenticator A, requestor R, and collaborator C), and the main ProVerif process considers arbitrarily many sessions of a protocol running concurrently.

2.3.2 Verification Setup

Each protocol was verified in 3 runs.

1. The first run began with attackers knowing no secret keys.

2. The second run began with attackers knowing all the long-term keys accessible to the collaborator. For the protocols shown in Figures 2.1–2.3, attackers were given K_{AC}, and for the protocol shown in Figure 2.4, attackers were given K_{AC} and K_{RC}.

3. The third run began with attackers knowing all the long-term keys accessible to the requestor. For the protocols shown in Figures 2.1–2.3, attackers were given K_{AR} and $\widehat{\widehat{K_{AR}}}$, and for the protocol shown in Figure 2.4, attackers were given K_{AR} and K_{RC}.

In all 3 runs of each of the 4 protocols, we attempted to verify the following security properties.

- P1 Secrecy of the session key: The session key K_{SK} is only known to the authenticator and requestor. This property subsumes forward secrecy of session keys in the third run of the private-channel protocols (Figures 2.1–2.3) because knowing the requestor's future authentication secret ($\widehat{\widehat{K_{AR}}}$, which becomes K_{AR} in the next round of authentication) does not leak session keys.

- P2 Authentication of R to A: With one exception, we specified authentication of R to A as requiring that if the authenticator receives an acknowledgment of a session key (and therefore believes it shares the session key with the requestor) then the requestor was indeed its interlocutor and the collaborator indeed collaborated. This is an event-based property [41] having the form

  $$endA(beginA \wedge collabA),$$

  where $endA$ refers to the event of A receiving the acknowledgment, $beginA$ to R sending the authentication request, and $collabA$ to C sending its participation message (in the third message of Figures 2.1 and 2.2 and the second message of Figures 2.3 and 2.4).

  The one exception to encoding *P2* in this way is for the second run of the all-public-channel protocol (Figure 2.4), where the attacker is given K_{AC} and K_{RC}. In this case, the attacker may use K_{RC} to obtain, and K_{AC} to collaborate with, legitimate authentication requests, thus helping legitimate authentications succeed, which we do not consider an attack. Therefore, for the second run of the Figure-2.4 protocol, we specify property *P2* as only requiring

  $$endAbeginA,$$

  that is, if the authenticator believes it shares the session key with the requestor then the requestor was indeed its interlocutor (but the attacker, rather than the collaborator, may have collaborated).

- P3 Authentication of A to R: This property is symmetric to *P2* and, with one exception, requires that if the requestor sends an acknowledgment of a session key (and therefore believes it shares the session key with the authenticator) then the authenticator was indeed its interlocutor and the collaborator indeed collaborated. This property has the form

$$endR(beginR \wedge collabR),$$

where *endR* refers to *R* sending the acknowledgment, *beginR* to *A* receiving the authentication request, and *collabR* to *C* sending its participation message.

As with *P2*, the one exception to encoding *P3* in this way is for the second run of the all-public-channel protocol (Figure 2.4), in which case *P3* only requires

$$endRbeginR,$$

for the same reason explained for property *P2*.

2.3.3 Verification Results

ProVerif found no attacks on any of properties *P1*–*P3* in any runs of any of the protocols. That is, ProVerif did not refute any of *P1*–*P3* in any runs of any of the protocols.

ProVerif did prove *P1* and *P3* for all 3 runs of all 4 protocols, and it proved *P2* for all 3 runs of the full coauthentication protocol (Figure 2.1). It also proved *P2* for the second and third runs of the protocol shown in Figure 2.4.

For all runs of the protocols shown in Figures 2.2 and 2.3, and for the first run of the protocol shown in Figure 2.4, ProVerif outputs that *P2* “cannot be proved”. It produces a trace in which a man-in-the-middle sits between *A* and *R*, and *A* and *C*, and simply collects and forwards all messages sent to and from *A*. This trace is not an attack because the authenticator completes the protocols with *R* having sent the original authentication request and *C* having sent its participation message, despite the fact that the attacker touched these messages while acting as an intermediary.

Table 2.1: Average performance of the authentication systems over 100 runs with A being the authenticator, R the requestor, and C the collaborator.

	Bytes	Application-Layer Time (ms)				Authentication
Implementation	Transmitted	A	R	C	Total	Time (ms)
Password	3212	0.28	1.5	—	1.8	136
Figure 2.1	1198	2.58	22.5	18.4	43.5	594
Figure 2.2	1088	1.36	20.1	20.9	42.4	475
Figure 2.3	885	1.16	17.2	19.4	37.8	473
Figure 2.4	1075	0.94	14.9	23.3	39.1	131

We also note that these results are for the stronger, injective-correspondence versions of properties *P2* and *P3*. The injective-correspondence versions require there to be a unique predecessor event for each end event [40, pp.19–22]; for example, the injective version of *P2* requires that for each *endA* event there exists a unique *beginA* predecessor event. The non-injective versions allow end events to have non-unique predecessor events. ProVerif was able to prove the weaker, non-injective version of property *P2* for all runs of all protocols.

2.4 Empirical Evaluation

We have implemented and measured the performance of full coauthentication (Figure 2.1) and the variations shown in Figures 2.2–2.4. To establish a baseline of performance, we also implemented and measured the performance of a basic password authentication system.

The authenticator in all implementations was the same MacBook Pro laptop, and the requestor and collaborator in all implementations were the same Android phones, except that the password-based implementation did not use a collaborator device. In all implementations, the authenticator, requestor, and collaborator processes ran as Java applications.

The password-based implementation communicated over HTTPS (using 2048-bit RSA and self-signed certificates), while the coauthentication implementations sent public-channel messages over TCP on standard Wi-Fi channels. Private-channel messages were sent through Bluetooth, though it has known vulnerabilities [42].

All symmetric cryptographic operations were implemented with 256-bit CBC-mode AES using the standard `javax.crypto` library, and all (64-bit) nonces, (256-bit) session keys, and (256-bit) updated-K_{AR} keys were dynamically generated using Java's cryptographically strong random number generator class `SecureRandom`. All other cryptographic keys were hardcoded, the initial shared keys being assumed to have been shared before the implementations began running. An (8-character) username and password were also hardcoded for the password-based implementation.

Each run of each implementation opened new network connections, including a new Bluetooth connection in the implementations of Figures 2.1–2.3. Connections were never reused between runs of the implementations, and the Android applications were restarted for each run.

Each of the implementations was run 100 times, in a uniform environment of normal (workday) university-network usage and standard loads of kernel and user-level applications running. The following measurements were made for each run:

- The network usage, that is, the number of bytes transmitted over the course of the run. Due to unreliability in the communication channels, the number of bytes transmitted varied with each run. The network usage was measured with Android's standard network-monitoring class `android.net.TrafficStats`.

- The application-layer real time each device consumed. This measurement was made by starting a timer when beginning to process any newly received message or request, stopping the timer when finished preparing a response, taking the difference, and summing all of these times for each device. For example, the application-layer real time consumed by the authenticator in full coauthentication is the sum of the real times it consumes processing the requestor's and collaborator's messages, including generating new keys and a challenge nonce and performing the required encryptions and decryptions. Application-layer times exclude all time spent establishing connections and transmitting messages in the underlying TCP, HTTPS, and Bluetooth protocols.

- The total authentication time. This is the real time, measured on the requestor, from beginning to prepare an authentication request until finishing obtaining a plaintext session key.

As shown in Table 2.1, the implementations transmitting more or more complex messages, or using HTTPS, transmitted more bytes of data. Network (i.e., non-application-layer) activities dominated the performance of all implementations, consuming between 70% and 98.7% of the total authentication time on average.

In terms of application-layer performance, the password system was the most efficient, benefiting (at the application layer) from pushing all the cryptographic operations into the underlying HTTPS layer.

In terms of total authentication time, the Figure-2.4 system outperformed the others on average. The performance of this coauthentication system benefits from transmitting a smaller number of messages over the efficient (relative to HTTPS and Bluetooth) TCP.

Importantly, these performance results exclude human time, though it is known to be substantial for password-based authentication systems. Human entry of a password is expected to take on the order of several seconds [14, 15, 43].

Care should also be exercised when comparing the performance of the password-based system with the performance of the private-channel coauthentication systems (Figures 2.1–2.3), which update K_{AR} on every authentication. The advantages of updating K_{AR} are analogous to the advantages of updating a password, so a better comparison might take into account the time required to update passwords. Password update is expected to take on the order of a minute of human time [44], significantly longer than an automatic coauthentication-key update.

We conclude from these results that coauthentication performs efficiently enough to be practical.

2.5 Extensions and Generalizations

Extensions and generalizations of coauthentication are possible.

2.5.1 Multiple Collaborators, m-out-of-n Policies, and Availability Benefits

There are advantages to systems in which users register more than two devices with an authenticator. Suppose a user has registered n devices and the authenticator requires any m of the n devices to coauthenticate, where $2 \leq m \leq n$. In the coauthentication protocols described so far, m=n=2, but now suppose m=2 and n=3. In this case, compromising only one of the user's devices (i.e., obtaining only one device's authentication secrets) is still insufficient for authenticating as that user, because m=2. At the same time, because m<n, the user can be authenticated even after forgetting or losing a device, or having a device become inoperable, for example due to a denial-of-service attack.

This m-out-of-n-device policy, enforced at the authenticator, tolerates the absence of $n-m$ devices. Hence, user-side denial-of-service attacks require denying service to $n-m+1$ devices. When these devices communicate through heterogeneous channels, denial-of-service attacks based on jamming or otherwise interfering with specific communication channels become more difficult to mount.

To prevent attackers from using $n-m$ compromised devices to coauthenticate, m may be further constrained to be greater than $n-m$, that is, $m > n/2$. For example, a system that requires only 2 out of 4 devices to coauthenticate (i.e., m=2=$n/2$) tolerates the absence of 2 devices, but if those 2 devices are absent due to theft, then the thief can use them to coauthenticate. To prevent such attacks, the m-out-of-n-device policy may be constrained to $2 \leq m \leq n < 2m$

The m-out-of-n-device policy can be generalized further, to policies in which devices are, for example, (1) weighted in various ways to get above a threshold (e.g., 2 “votes” are required to authenticate the current user, but each smart shoe only gets half a vote), (2) required (e.g., 2 devices are required but one must be the user’s smartphone), or (3) excluded (e.g., high-risk users may not use easily-transferrable smartcards for coauthentication).

2.5.2 Group Coauthentication

Users may also be coauthenticated simultaneously, as a group. Such authentication subsumes the famous two-person concept for authenticating users who will have access to nuclear and other weapons [45, 46], or to bank vaults. For example, a two-person policy may require two users to simultaneously turn four keys, one in each hand, to gain access to a weapon-deployment system. The goal is to require both users to participate in the authentication.

Because coauthentication requires participation of multiple devices in an authentication, it may require participation of multiple users in an authentication, where each user has at least one registered device. The same coauthentication protocols can be followed to authenticate multiple users’ devices simultaneously. More sophisticated group coauthentications could, for example, require participation of m-out-of-n devices from each of j-out-of-k users.

2.5.3 Device Sharing and Anonymous Coauthentication

Users may also share devices. For example, a garage-door authenticator may receive a request from a shared family car and send challenges to all the smartphones of drivers in the family, or only those smartphones in near-proximity. The smartphones might enforce the collaboration policy of tacitly participating if co-located with the requestor and not participating otherwise.

Alternatively, assume that every collaborator (smartphone) shares *the same* secret key with the garage-door authenticator. Then the authenticator may, upon receiving a request from the family car, respond directly to the car with a challenge requiring participation from any collaborator—and leave it to the car to obtain a collaborator's participation. An interesting aspect of this alternative is the anonymity it provides: the authenticator only communicates with the shared requestor device and does not know which user has been authenticated, nor which device has collaborated. Authentications are still protected against attackers acquiring one of the secret keys.

It is also possible to achieve anonymous coauthentication for systems in which requestor devices are not shared, by having all potential requestors share the same secret key with the authenticator. Because coauthenticators verify usage of keys, anonymity is achieved by having devices share keys.

Of course, these designs only protect anonymity during the authentication process. Authenticators frequently have other opportunities to de-anonymize users, though techniques like onion routing [47] may mitigate some de-anonymizations.

2.6 Additional Discussion of Related Work

Many existing systems are related to coauthentication.

2.6.1 Threshold Schemes and Multi-Signatures

An (m, n) threshold scheme enables a secret to be divided among n entities, such that each entity has one piece of the secret and m of the n pieces are required to determine the secret [48]. An (m, n) threshold scheme has cryptographic benefits analogous to the user-authentication benefits of an m-out-of-n-device coauthentication policy; both protect against fewer-than-m entities acting maliciously and at-most-n-minus-m entities being unavailable to participate.

Multi-signature schemes similarly enable different users or devices to generate a joint digital signature [49].

Threshold and multi-signature schemes do not provide coauthentication systems, and vice versa, as they differ in techniques and goals. Threshold (multi-signature) schemes contribute techniques for combining secret-pieces (signatures) into a joint secret (signature), while coauthentication systems require no joint secret or signature. Coauthentication secrets (i.e., keys) may be used only independently, to indicate one device's participation in user-level authentications, without ever being combined. The goals of threshold and multi-signature schemes focus on combining pieces of cryptographic secrets or signatures into joint secrets or signatures, while coauthentication's goals focus on user authentication.

2.6.2 OTPs and Other Techniques Using Multiple Devices

One group of techniques related to coauthentication uses OTPs, as discussed in Section 1.1. The standard use of OTPs is as follows. A user enters a username and password on a requestor device, the authenticator SMS-texts an OTP to the user's phone (which may also be the requestor device), and the user sees the OTP and enters it on the requestor device as a second password required for authentication. This use of OTPs differs from coauthentication in several ways, perhaps the most significant being that the OTPs are used in two-factor systems, while coauthentication is a single-factor system. Hence, attackers can break the OTP portion of authentications by compromising one device, the victim's phone, or by reading the SMS messages sent to the phone [9, 50, 51].

Another related group of techniques use multiple devices to acquire multiple passwords or biometric data [52]. The authenticator combines these data to determine whether to authenticate a user. For example, if a user has a sensor-device implanted in each finger, then each device may send data related to that finger's motion to an authenticator, which can make authentication decisions based on whether a user has moved or gestured in the proper way for that user. Although using multiple devices, this line of work relies on users to enter passwords or biometrics, which are assumed to be unguessable and unforgeable by attackers.

Coauthentication, like other zero- or low-interaction authentication systems [36], shields users from attacks based on guessing or forging authentication secrets, such as password phishing or biometric surveillance. Coauthentication users never have to access or even understand the secrets required for authentication, and coauthentication secrets can be generated automatically, with high entropy, and without concern for whether humans have the resources (cognitive ability, time, etc.) to generate, store, update, or enter the secrets.

Bonneau et al. evaluated authentication techniques, including OTPs, according to three axes: usability, deployability, and security [53]. A total of 25 criteria are considered along these axes, such as whether the techniques require users to memorize secrets or carry devices. As motivated in Section 1, we consider disadvantages related to requiring users to carry devices to be decreasing. In any case, we believe that coauthentication satisfies the majority of Bonneau et al.'s criteria, though it is difficult to make precise claims in this respect, due to subjectivity in the criteria [53, Section V-B]. The most significant criteria coauthentication does not satisfy relate to deployability; deploying coauthentication, like deploying any new authentication technique, would require updating authentication clients and servers, and in some implementations, relying on co-location verification.

2.6.3 Using Coauthentication Protocols to Implement Existing Multi-Device Techniques

Coauthentication protocols can be used to implement existing multi-device authentication systems.

For example, the full coauthentication protocol shown in Figure 2.1 can implement OTP-based authentication: the requestor might be a laptop, the initial (static) password might be included in the initial authentication-request message sent from requestor to authenticator, the challenge nonce might be the one-time (dynamic) password, the collaborator might be a smartphone, the communication from authenticator to collaborator might be through SMS, and the communication from collaborator to requestor might occur by displaying the OTP-carrying ciphertext on the smartphone screen and having the user enter it manually on the laptop.

Similarly, the protocol shown in Figure 2.2 can implement authentication based on biometric data collected from multiple sensors, for example, authentication based on data collected from sensors implanted in fingers [52]. In this case the requestor may request collaboration from multiple sensors, each of which transmits its authentication participation—including motion data collected—to the authenticator. The authenticator collects and considers these participation messages to make authentication decisions.

Implementing existing multi-device authentication systems with coauthentication protocols provides the formally verified security benefits outlined in Section 2.3. These benefits are sometimes lacking in the existing systems. For example, the protocol shown in Figure 2.1 provides forward-secrecy properties lacking in many existing authentication systems. In addition, although existing OTP systems are vulnerable to text-message eavesdropping and man-in-the-middle attacks [9, 50, 51], the coauthentication protocols mitigate these attacks.

Chapter 3: Usability Experiment Methodology

This chapter details the study (IRB-approved, see Appendix D) methodology according to the framework's principles. The study was conducted in a lab where each participant followed instructions from a web application on a laptop. A researcher also guided them through the procedure and answered questions. This section will present the authentication techniques evaluated; then discuss the recruitment process and the resulting demographics; then we will detail what participants underwent, the specific hardware used, and finally the limitations associated with such a study.

3.1 Authentication Techniques Evaluated

The authentication techniques evaluated in this paper's framework represent the three main authentication factors (i.e., knowledge, inherence, possession). Passwords were chosen to represent the knowledge factor because of their popularity. A fingerprint authentication method was chosen to represent the inherence factor, because of the availability of fingerprint sensors on mobile devices, and because fingerprint authentication has been well researched [54, 55, 56]. Coauthentication was chosen to represent the possession factor because this authentication method has never been evaluated [1].

The coauthentication and password authentication methods were completed on two different input devices: a laptop and a phone. Therefore, including fingerprint authentication on phone, a total of five authentication techniques were evaluated. Testing on two devices allowed for comparison between input devices.

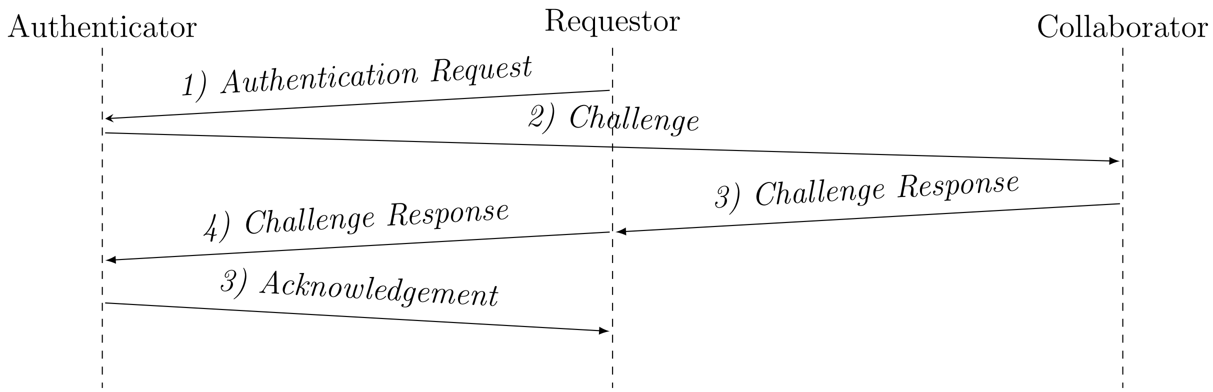


Figure 3.1: Overview of the full coauthentication protocol [1, Section 3].

Each of these authentication methods requires a user registration prior to authenticate, in order to register a password, fingerprint, or device keys (cryptographic secrets used for coauthentication). Thus, the registration phase was done prior to the start of the experiment.

3.1.1 Fingerprint

The fingerprint authentication method evaluated uses Android's built-in fingerprint features, and the user is authenticated locally. No fingerprint or fingerprint hash is transmitted over the network. Due to the sensitive nature of fingerprints and the Android security features, only a confirmation message is sent to the server to indicate the authentication's result. Android stores the hashes of the fingerprint in a trusted execution environment [57].

In this study, only one authentication attempt was accepted, and only one fingerprint was registered. However, Android policy allows multiple authentication attempts to increase usability. The reason for a single attempt is to provide standardized accuracy among all authentication methods compared. In this experiment, none of the authentication methods allowed for retries.

3.1.2 Password on Laptop

The password method follows modern standards, using the Password-Based Key Derivation Function 2 [58, 59] to hash the password with the following arguments: 1000 iterations, 256-bit length, and a salt, all of which are recommended values [60]. The password was given to the participants, and they had to type it while the username was pre-filled. The password, "MxmwS88V" is 8 characters long with upper case letters, lower case letters, and numbers, and therefore satisfies standard policies [61, 43, 62].

These design choices were made for multiple reasons:

1. A set password allows for comparing the typing speeds between participants.
2. Allowing participants to choose their passwords would allow them to create simple passwords (i.e, easy to type) that meet the minimum requirements in order to maximize their compensation.
3. Numerous web services keep track of users and pre-fill their usernames.

During the experiment participants were instructed to use only alphanumeric keys to prevent them from copying and pasting the password.

3.1.3 Password on Phone

This method is included to compare usability results between input devices. The password authentication technique on phone is the same as password on laptop. The main difference is that participants have to type the password on the phone's virtual keyboard. The password is the same in both methods, and the username is also pre-filled.

3.1.4 Coauthentication on Laptop

Coauthentication is a new authentication method using the possession factor [1]. This authentication method is single factor, but multi-device, and its usability has never been evaluated.

Coauthentication can be implemented with various protocols, each having advantages and disadvantages. The full coauthentication protocol is the most likely to be implemented, so it was an ideal candidate to be evaluated. To authenticate, users need two two registered devices, which will collaborate to perform a challenge-response security protocol.

Figure 3.1 illustrates the underlying protocol of coauthentication, with the requestor being the laptop and the collaborator being the phone. The implementation of the protocol uses symmetric keys to encrypt messages with the standard AES cipher in CBC made with PKCS5Padding.

3.1.5 Coauthentication on Phone

This method is included to compare usability results between the phone and laptop. It follows the same protocol illustrated in Figure 3.1. However, for this authentication technique the requestor is the phone and the collaborator is the laptop.

3.2 Related Work of Authentication Usability Methodology

3.2.1 Usability Metrics

The three main aspects of usability (efficiency, effectiveness, and satisfaction) should be considered to properly assess a system's usability. Indeed, these aspects are not always correlated, and assumptions on the overall system's usability may not be accurate [63, 64, 65].

The satisfaction aspect of an authentication method is often the main research focus as the adoption of an authentication method depends on end-users. The System Usability Scale

(SUS) is considered a standard to collect this satisfaction measurement via a study collecting participants' perceptions [17, 29, 18, 66, 2]. The questions from the SUS questionnaire are answered via a Likert scale that allows for the calculation of a usability score from 0 to 100 [19, 67]. This score has been assessed for a wide variety of computer systems and revealed to be consistent and reliable [68, 69]. SUS is also a recommended metric standard to compare authentication methods [18].

The review of the following questionnaires, has deemed SUS as the most appropriate questionnaire to assess [68]:

1. Questionnaire for User Interface Satisfaction (QUIS) [70],
2. System Usability Sacale (SUS) [17],
3. Computer System Usability Questionnaire (CSUQ) [71], and
4. Microsoft's Product Reaction Cards (MPRC) [72],

The findings show that the data collected via the SUS questionnaire yield the most reliable result. However, the authors point out the fact that more data, with a greater number of participants, needs to be conducted to correlate their findings. A crucial point to mention expressed by the author is the versatility of the SUS questionnaire which focus on a whole system and not on specific points. This is the main reason to use SUS over other questionnaires for authentication method usability assessment. Indeed, all questions from the SUS questionnaire refer to "the system" which can be replaced by "authentication method" and retain meaning. Other questionnaire do not allow for such simple adaptation to which system is assessed.

The efficiency aspect of usability is more straightforward than the satisfaction aspect as it can simply be calculated by recording the time to complete tasks. In term of authenticating the time it takes for user to authenticate can represent the efficiency of an authentication method. The completion time should be calculated from the first user action to the result of the authentication. This would be a fair assessment of an authentication time even for implicit authentication which determine a user's credential by analyzing their actions [73].

The effectiveness aspect of usability depends on a user's ability to perform and achieve specific goals. In term of authentication the false positive rate and false negative rate indicates the effectiveness. Indeed, an authentication is not effective if it refuses access to a legitimate user or grants access to an illegitimate user.

Usability is still a field in full expansion especially due to the emergence of new platform and system with different requirement from a standard personal computer or laptop. Additional aspect have been proposed in the past such as: 1. Fun [74] 2. Aesthetics [75] 3. Sociability [76] 4. Flow [77] 5. Learnability [78] . However, these aspects have not been agreed upon, can be considered a subset of another aspect, or are specific to the system assessed.

In Human Computer Interaction usability is still argued on how to properly measured it and constant improvement is made in this field. Therefore, security researchers focusing on authentication need to take into account new findings and technique that could be appropriate and related.

The efficiency and effectiveness usability aspect are appropriate and relatively straightforward to assess when pertaining to authentication because of their quantifiability. However, the satisfaction aspect, while still being appropriate, can be measured in various ways. More research with the System Usability Scale (SUS) assess on authentication needs to be proposed; to either develop a novel questionnaire more appropriate or conclude that the SUS questionnaire is sufficient and/or appropriate enough.

3.2.2 Authentication Methods Subjective Overall Comparability Methodology

Comparing authentication methods is often achieved by comparing methods as a whole, which results in a high-level overview of usability [79, 3].

3.2.3 The Quest to Replace Passwords Scale

The Quest to Replace Passwords [3] provides a qualitative scale to compare authentication and encapsulates the three main aspects of usability: efficiency, effectiveness, and satisfaction. Table 3.1 illustrate each criterion and their relation to the usability aspects. However, this scale is difficult to assess objectively to compare authentication methods, due to the subjective nature of the criteria [3, Section V-B].

Table 3.1: Quest To Replace Password usability criteria categorized depending on their corresponding usability aspect. A "∼" means that the criteria can be considered part of the aspect while a "✓" means it is and a "×" means it is not.

QTRP usability criteria	Usability aspect		
	Satisfaction	Efficiency	Effectiveness
U1: Memorywise-Effortless	∼	×	×
U2: Scalable-for-Users	∼	×	×
U3: Nothing-to-Carry	∼	×	×
U4: Effortless	∼	∼	×
U5: Easy-to-Learn	∼	∼	×
U6: Efficient-to-Use	∼	✓	×
U7: Infrequent-Errors	∼	×	✓
U8: Easy-Recovery-from-Loss	×	×	×

The following is the description of each of the usability rating illustrated in Table 3.1 as described by the original author [3]:

- U1 *Memorywise-Effortless*: Users of the scheme do not have to remember any secrets at all. We grant a Quasi-Memorywise-Effortless if users have to remember one secret for everything (as opposed to one per verifier).

- U2 *Scalable-for-Users*: Using the scheme for hundreds of accounts does not increase the burden on the user. As the mnemonic suggests, we mean "scalable" only from the user's perspective, looking at the cognitive load, not from a system deployment perspective, looking at allocation of technical resources.

- U3 *Nothing-to-Carry*: Users do not need to carry an additional physical object (electronic device, mechanical key, piece of paper) to use the scheme.Quasi-Nothing-to-Carryis awarded if the object is one that they'd carry everywhere all the time anyway, such as their mobile phone, but not if it's their computer (including tablets).

- U4 *Physically-Effortless*: The authentication process does not require physical (as opposed to cognitive) user effort beyond, say, pressing a button. Schemes that don't offer this benefit include those that require typing, scribbling or performing a set of motions. We grant Quasi-Physically-Effortless if the user's effort is limited to speaking, on the basis that even illiterate people find that natural to do.

- U5 *Easy-to-Learn*: Users who don't know the scheme can figure it out and learn it without too much trouble, and then easily recall how to use it.

- U6 *Efficient-to-Use*: The time the user must spend for each authentication is acceptably short. The time required for setting up a new association with a verifier, although possibly longer than that for authentication, is also reasonable.

- U7 *Infrequent-Errors*: The task that users must perform to log in usually succeeds when performed by a legitimate and honest user. In other words,the scheme isn't so hard to use or unreliable that genuine users are routinely rejected.

- U8 *Easy-Recovery-from-Loss*: A user can conveniently regain the ability to authenticate if the token is lost or the credentials forgotten. This combines usability aspects such as: low latency before restored ability; low user inconvenience in recovery (e.g.,no requirement for physically standing in line);and assurance that recovery will be possible, for example via built-in backups or secondary recovery schemes. If recovery requires some form of re-enrollment, this benefit rates its convenience.

Satisfaction is a highly subjective aspect of usability, therefore relaying on a single or small group of individuals to assess a qualitative scale, can provide an estimation of the satisfaction aspect. However, such estimate can be involuntarily biased. For example, the U8 criteria *Easy-Recovery-from-Loss* is imprecise on how to define the easiness. One could argue that recovering by making a phone call is easy while other considering a phone call cumbersome. In addition, such criteria does not allow for comparison between recovering techniques. For example, having to send an email versus making a phone call. Examples of this impreciseness can be thought through for other criteria, indeed two different authentication method can be assessed as *Infrequent-Errors* but one may have a higher error frequency than the other.

Additionally, this coarse grained issue of subjective qualitativeness also arise for the efficiency and effectiveness aspects. Indeed, giving a qualitative rating for these quantitative aspects does not allow for a precise comparison.

Another specificity of this scale worthy to mention is also the scope of it. The scale aims to be broader than solely the authentication process. Indeed, the U8 *Easy-Recovery-from-Loss* criteria coverage is definitely important as it is the main drawback of some authentication methods. However, the loss or theft of the authentication secret can be argued to be separate from the authentication process itself. If recovery in case of loss or theft is a factor taking into account the registration phase of an authentication method could also be included. Indeed, some registration phase can be quite cumbersome or difficult to achieve for some authentication schemes.

3.2.4 Applicability of Usability Principle to Security Systems

Braz et al., methodology to compare authentication method consist in subjectively describing if the 8 golden rules [80] of user interface design apply to authentication technique. These golden rules are:

1. Strive for consistency

2. Frequent users can use shortcuts

3. Provide informative feedback

4. Dialogs should yield closure

5. Prevent errors and provide simple error handling

6. Easy reversal of any action

7. Put the user in charge

8. Reduce short-term memory load

The authors pointed out the inadequacy of these rules in the case of user authentication and provide descriptions of each of the authentication techniques studied. These descriptions give a high level understanding of the usability of each of the authentication techniques studied. However, a subjective description cannot provide a precise comparison across the spectrum of authentication methods.

The author illustrates a comparison between each of the authentication techniques which, for usability, only provide the number of golden rule followed. These criteria likely aims to evaluate the satisfaction aspect of usability. Indeed, the comparison contains separate criteria for the efficiency and effectiveness aspects.

Assessing the usability of an authentication method subjectively allows for a rough estimate but fail to provide a precise comparison or a comparison at all.

3.2.5 Usability Methodology for Passwords

As is well known, passwords are the predominant authentication method, and an extensive literature has evaluated their usability over the last few decades. However, most usability research specific to passwords is not directly applicable to other authentication methods. The research mostly pertains to improving the usability of passwords.

In these studies the experimental setup is similar where participants have to type passwords and answer questionnaires to collect feedback, demographic data, and often typing times. Depending on the goal of the study, participants are given the password(s) or are required to create a password(s).

3.2.6 Password Policies Usability

Recent studies assessed on passwords often focus on the effects of password policies on usability [43, 81, 82]. Studies usually measure the annoyance of password policies and often the resulting user behavior.

Many password policies are often enforced without taking into consideration the usability induced. The security and usability of many combinations of password creation policies have been compare to find the most adequate in term of security benefit and usability [43].

Password expiration policies are becoming a new standard practice in Universities and many organizations. This practice can be thought to provide the same benefit than the principle of key update. However, the research on password policies [82, 83, 84, 62], shows that user compelled to create new passwords often do so by adding a simple modification to their previous password ($\sim 75\%$ of participants). Zhang et al. [83], were able to crack a large amount of their organization's password by using the previous password used.

The methodology employed in password usability studies are either: 1. reasoning on a dataset of passwords which can help draw conclusion on user behavior 2. or subjecting participants to the policy(ies) of interest and gather data on participant's opinion in addition to timing data and the passwords themselves. In this case the study can often be more thorough because research have the opinion of participants but the data is considered not fully ecologically sound compared to studying "in-use" passwords.

3.2.7 Password Device Entry Usability

Another focus of recent studies is the typing speed of users due to the emergence of smartphones and the needs to enter passwords on virtual keyboards [14, 85].

In these studies participants are required to construct and/or enter passwords in different type of keyboards. The data collected and the analysis resulting focuses on the efficiency and eventually effectiveness of the different keyboard evaluated.

The methodologies employed are not well suited for other authentication technique due to their specificity. Indeed, in the example of typing speed some research incorporate effectiveness by investigating the error rate induced by some keyboard but the focus is made mainly on the efficiency aspect of usability.

3.2.8 Usability Methodology for Biometrics

The focus of biometric usability research is similar to the research on password usability in that a biometric method's usability is typically only compared to other biometric authentication methods.

For biometric methods the action required for a user to be authenticated, is usually done by simply scanning the biometric featured required. Therefore, the satisfaction aspect of biometric is not the main concern. Biometric methods are comparable via a False Acceptance Rate (FAR) and a False Rejection Rate (FRR) [86, 87, 88, 28]. The FAR is often used to indicate a level of security while the FRR is often used to indicate a level of usability. These clearly defined metrics are the reason why biometric methods are easily comparable.

However, non-biometric authentication technique do not always allow for the calculation of such metric and therefore cannot be directly compared.

The FRR rates indicate the effectiveness of the authentication methods but does not give any indication on the efficiency or satisfaction of an authentication technique. The efficiency aspect is important and depend on the hardware used.

Since effectiveness and satisfaction are not correlated [64, Section 4.5], effectiveness is insufficient to determining a usability that can be compared to the usability of authentication methods using other authentication factor(s).

The efficiency and effectiveness of a biometric authentication technique also depends on its application. Indeed, if the authenticator, the entity determining the user's identity, has to match the biometric feature with a single authorize user or among a group of user the efficiency and effectiveness of the technique will be affected.

For the more established biometric technique such as fingerprint, or face recognition, large datasets are available to test and determine the FAR and FRR rates. So the methodology for obtaining a FRR and a FAR rate is to use those datasets.

3.3 Study Recruitment and Demographics

Participants for this study were recruited in various ways. Primarily, the cloud-based participant pool management software SONA Systems was used through the University of Florida's Psychology department to recruit students enrolled in the general psychology course. Students taking the course were required to sign up for studies, and received 4 credits for participation in addition to the extra compensation based on task performance. Additional participants were recruited using flyers.

Table 3.2 shows the demographics of the 43 participants enrolled in the study. All interested participants were accepted; however, the recruitment methods attracted primarily

Table 3.2: Demographics of the 43 participants enrolled in the study.

Demographic Category	# Participants ($N = 43$)	Percentage
Gender		
Male	10	28%
Female	33	72%
Age		
18 years old	14	34%
19 years old	16	36%
20 years old	9	23%
21 years old	4	7%
Ethnicity and Race		
Hispanic or Latino	9	20%
Black	4	9%
Asian	18	40%
White	28	64%
Language		
Bilingual	20	45%
Native English	36	82%

college students (all under 22 years old). Additionally, 33 out of the 43 total participants were female, so this study has a disproportionate representation of the female demographic.

3.4 Study Design

According to the framework guidelines, described previously in Section 1.3, the participants completed the following steps:

1. The Participant Information Questionnaire
2. A training phase for the authentication techniques (i.e., standalone)
3. A training phase for the user activity
4. The Authentication Experience Questionnaire to collect data on authentication alone
5. A training phase for the Dual-Task Interference game (i.e., DTI game)

6. The DTI game (Administered in six sessions)
7. The Authentication Experience Questionnaire a second time to collect data on authenticating while multi-tasking

For this study we chose to make the user activity simulate a conversation, to represent the common use case of authentication interrupting conversation. To mimic a conversation, participants repeated a series of words. The accuracy of correctly repeating these words was recorded, and participants acquired two points for each correct word. Each conversation lasted five minutes and used the same series of words. There were a total of six conversations, for a total of thirty minutes. For the remainder of this paper, these conversations will be described as the DTI game or multi-tasking game.

Participants also accumulated more points by successfully performing the authentications required. Each successful authentication earned ten points. In order for the participants to know each authentication result, it was displayed for two seconds.

To get familiarized with the various components of the experiment, participants went through three training phases. In the first training phase, participants repeated a series of words for thirty seconds to simulate a conversation. The second phase required the participants to perform each of the authentication methods twice. In the third training phase participants had to repeat words while authenticating for one and a half minutes (i.e., practice the multi-tasking game).

To collect participants' feedback, the Authentication Experience Questionnaire (AEQ), was given twice. The questionnaire was given once after participants performed the first training phase (i.e., authentication methods training), and a second time after the multi-tasking game. This repetition allows for comparison between authentication performed in a standalone manner versus during the DTI game.

The participant's score was the only incentive for participants to perform the experiment properly. The score was updated and displayed in real time during the game. After each conversation of the game the participant could see the score earned and take a break. At the end of the experiment the compensation was calculated from the best score obtained between all six games. Each successful authentication earned 10 points and each successful audio task earned 2 points.

3.5 Hardware Specification

To complete the experiment, participants used a laptop and a smart-phone provided. The server (i.e., authenticator) was also deployed on the same laptop. The following is the hardware specification of these devices:

- Laptop: Dell Windows 10, memory 8GB, processor i5-7200U 2 cores at 2.5 GHz, and a 13 inch screen size.
- Phone: LG V20 with 4GB of memory, a 1.6GHz quad-core processor, Android 7, and a 5.7 inch screen size.

One relevant specification of the hardware here is the placement of the fingerprint scanner, which was located on the back of the smart-phone.

3.6 Limitations

The demographic data shows that most participants were female (73%), college students (100%), and relatively young (100% are under 22 years old). Thus, the data obtained is more useful at predicting usability in female college students than any other group.

Technologically, each individual's comfort with the specific hardware aspects of the experiment can be considered a confounding variable. Many of our participants use Apple products such as the iPhone and Mac computer. The level of comfort these participants had with our Android phone and Windows computer may not match that which they typically feel for their personal devices. For example, the screen size of both the laptop and the phone may differ from the ones the participants are used to. Additionally the collection of each participant's response to an authentication task was contingent on the processing power of the hardware used. This contingency can create a confounding variable related to the quality of the devices used in the study.

The audio component of the framework meant to simulate a conversation being had while authentication was simultaneously performed may also represent a limitation for our framework. Conversations often involve more than simply repeating words. Indeed, while this task still adequately serves as a second task, demanding at least some level of attention and inducing a multitask response from participants, its comparability to real life conversations is not optimal.

Gamification has limitations in terms of accustomization and age of participants. Therefore, the study should not be assessed multiple times for the same participant. The demographic data shows that the participants were relatively young, thus the age of the participants was not a concern in this regard. 19 years old 16 36%

Chapter 4: Results and Analysis

This chapter presents a quantitative and qualitative analysis of the data collected during the experiment.

4.1 Efficiency: Completion Time

The completion times are calculated from the start of the first user action to the reception of the authentication result.

Table 4.1 details the completion times for both the practice and multi-tasking game. Authentication tasks appear to participants in a random order. This design decision resulted in a similar number of authentication methods per participant. Therefore, the averages are weighted per participant.

Most completion times improved from the practice to the multi-tasking game, which can presumably be a result of participants' accustomization.

Table 4.1: Comparison of completion times for each authentication technique evaluated.

	Completion times (seconds)			
	Standalone		Multi-tasking game	
Authentication methods	Average	Median	Average	Median
Fingerprint	3.25	2.28	1.50	1.17
Password (laptop)	8.83	8.12	5.96	5.13
Password (phone)	9.25	8.75	6.78	4.99
Coauthentication (laptop)	0.68	0.63	0.74	0.49
Coauthentication (phone)	1.09	0.92	0.83	0.61

Table 4.2: Success rates for fingerprint and password authentication techniques weighted per participants.

	Authentication Success Rate (%)	
Authentication method	Standalone	Multi-tasking
Fingerprint	99.95	99.99
Password (laptop)	99.92	99.95
Password (phone)	99.86	99.95

4.2 Effectiveness: Success Rate

The success rate provides a metric for authentication effectiveness, which is determined by participants successfully initiating the authentication process and the reception of a successful authentication result. Table 4.2 shows the success rate for both the practice and the multi-tasking game.

Table 4.2 does not include coauthentication because, in a controlled environment (e.g., the elimination of network problems), coauthentication could not fail. In a more practical scenario network problems may be inevitable and coauthentication may fail. To ensure the completion of the experiment, the network had to be stable, thus coauthentication was not impacted by potential network issues.

The experiment was designed for participants to become well accustomed to the various tasks required and is the reason for such high accuracy. The authentication task training is not timed and is meant to be successful for the participant to understand what is required to be performed by each authentication method. Additionally, the data collected indicates that the success rate increased throughout the experiment. Indeed, during the multi-tasking game, more than 60% of failed authentication attempts appeared in the first two conversations (i.e., the first 10 minutes).

An important point about the fingerprint scanner success rate is that Android's policy requires multiple scans of a fingerprint to register a user, which increases the chance of success. Additionally, there was only one registered user.

Table 4.3: Averages of System Usability Scale scores of the authentication methods evaluated.

	SUS Score	
Authentication method	Standalone	Multi-tasking
Fingerprint	88	82
Password (laptop)	81	76
Password (phone)	78	74
Coauthentication (laptop)	81	82
Coauthentication (phone)	81	82

4.3 Satisfaction: Subjective Usability

The System Usability Scale (SUS) was used to measure the satisfaction of the participants. The SUS questionnaire was assessed within the Authentication Experience Questionnaire two times, a first time after the practice of the authentication task performed in a standalone manner and a second time after the DTI game. These SUS scores are shown in Table 4.3.

Fingerprint is highly rated in both standalone and during the multi-tasking game. The high score of the fingerprint authentication during the standalone portion can be explained by the high percentage of participants currently using fingerprint authentication in their daily life. Indeed, 72% of the participants enrolled stated to be using fingerprint authentication.

Coauthentication has an important improvement, from standalone to multi-tasking, which we believe is due to the novelty of this authentication method.

Password on phone's low SUS score is likely a result of the increased difficulty to type on virtual keyboards [85].

4.4 Additional Results

The accuracy of the words repeated was collected to identify difficult words for future experiments. Table 4.4 shows the accuracy only for words under ninety percent accuracy.

Table 4.4: Accuracy of words repeated (only for words under 90% accuracy).

Words to repeat	accuracy (in %)
fuss	34
lot	63
pat	75
keep	76
pop	79
sew	83
pay	88
sheep	89

Low accuracy rates can result from several factors: 1. The length of the word. Shorter words can be misheard or confused with other similar words. 2. The uncommonality of the words. 3. The non-native English speaker may have more trouble with pronunciation. In our study we have 18% of non-native speakers. 4. The audio may not be clear enough on some words.

The word "fuss" was correctly repeated only $\sim$ 34% of the time, which can be a result of a combination of these factors.

Since authentication requirement points are purposely weighted higher than the audio tasks (10 points versus 2 points), participants may choose to not repeat words to successfully perform an authentication task, thus accumulating more points.

Figure 4.1, shows the participants enrolled in the study that use each authentication method in their daily lives. The Technology and Authentication Experience questionnaire, included in the Participant Information Questionnaire (see Appendix A), makes a clear distinction between long and short passwords. A short password is explained as at most 6 typed characters, or passwords based on swiping a particular pattern, or making some sort of gesture, including passwords and PINs to log into phones, ATM, etc., while a long password is explained as at least 7 typed characters.

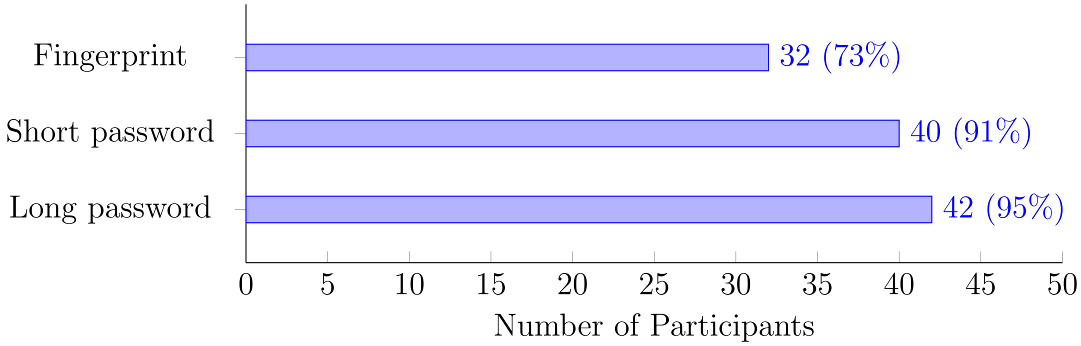


Figure 4.1: Percentage of participants that use each authentication method in their daily lives.

Chapter 5: Conclusion

5.1 Authentication Usability Methodology

Researchers mostly focus on improving current authentication techniques' usability by focusing within a specific type of authentication. For example, in the case of passwords, various composition policies (i.e. combination requirement for strengthening passwords) are required by different systems which lead researchers to focus specifically on studying the usability of password combination policies [43]. This specificity is not amenable to other types of authentication techniques. In biometric authentication methods, the necessary action for a user's authentication usually involves the required biometric feature to be scanned. Therefore the satisfaction aspect of biometrics is not the main concern. However, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are metrics used to compare one technique to another. These metrics represent the accuracy of such methods, but FAR is usually not applicable for non-biometric authentication techniques. In the case of security tokens, their usability is not well studied. This is a likely result of a more complicated experimental setup for such study.

All of the methodologies described in Section 3.2.1 are appropriate for the goals of their respective papers. However, such methodologies are often not applicable for different type of techniques.

The results of the study described in Chapter 4 show that fingerprint and coauthentication (both laptop and phone) are the more usable techniques evaluated. Their satisfaction and efficiency results are significantly better than passwords, though we are unable to draw conclusions regarding the effectiveness due to the similarity in results. Coauthentication yielded higher efficiency results than fingerprint. However, the satisfaction results of fingerprint are overall better or as good as coauthentication (both laptop and phone).

The framework developed enables the uniform evaluation of authentication methods' usability by using a standard methodology across various authentication factors, by focusing on efficiency, effectiveness, and satisfaction.

5.2 Coauthentication

The coauthentication protocols and system designs have several potential benefits. Coauthentication:

- protects against compromise of any one authentication secret, similar to multi-factor techniques but without the inconveniences of having to enter passwords (including OTPs) or scan biometrics;
- requires little, and in some implementations no, interaction from users;
- mitigates phishing, replay, and man-in-the-middle attacks (there are no passwords to phish, and the attack models assume active attackers);
- bases authentications on high-entropy secrets that can be generated, exchanged, stored, updated, and used automatically and efficiently (in contrast with password and biometric secrets);
- can implement advanced functionalities, including m-out-of-n, continuous, group, shared-device, and anonymous authentications;
- has formally verified security properties;
- has been implemented and found to perform efficiently enough to be practical;

- can be combined with additional authentication factors; and
- provides protocols that may benefit existing multi-device authentication systems, such as those based on OTPs.

Another benefit of coauthentication is its ability to reset secrets automatically. With existing systems, if a nonuser does obtain the required secrets, then resetting the secrets is laborious (e.g., for the victim to reset a password), expensive (e.g., to send the user a new physical token), or impractical (e.g., to give a user new fingerprints, retinas, vocal profile, etc). In contrast, coauthentication secrets may be cryptographic keys stored on registered devices; these keys may be reset, and periodically updated, automatically. These keys can also be generated to have high entropy, without concern for whether users can create, memorize, or enter the high-entropy secrets.

Because users never enter coauthentication secrets, these secrets cannot be phished by convincing users to enter them. In contrast, passwords are often obtained by convincing users to enter them as part of phishing attacks [12].

By providing an inexpensive, easy-to-use, and easy-to-deploy system of continuous authentication, coauthentication may enable applications like the following. Suppose that a number of terminals are distributed throughout a medical-care facility. The terminals actively poll nearby devices (e.g., through NFC or Bluetooth) for a new user, coauthenticate when one is found (without user interaction), and then display data allowable for that coauthenticated user with continuous coauthentication occurring (e.g., with a new coauthentication of the user every 20s). Then a physician might walk up to a terminal and without interaction or inconvenience—only by virtue of having his or her registered devices, like a smart ring, smart phone, and smart shoes—be able to see the current patient's medical records. When the physician leaves the terminal's proximity, the terminal may go blank and await the next user. If the next user is a coauthenticated station-prep attendant, then data specific for that user may be displayed.

5.3 Future Work

5.3.1 Coauthentication

Coauthentication can be implemented with various protocols and policies and with little-to-no user interaction. We hypothesize that coauthentication mechanisms may have improved usability when compared to existing multi-factor mechanisms. We presented a framework showing the usability advantages of two coauthentication implementations. This framework gave a promising evaluation of multiple authentication techniques. Coauthentication could be compared to some multi-factor implementations following the same methodology.

Coauthentication can be implemented for various applications and allows the integration of various features in the protocol. The following is a non-exhaustive list of additional modifications to coauthentication:

- Shared Association of Devices: In some implementations, a device could be shared between two users. For example, a device can be shared between both Alice and Bob. In this case, an insider attack could occur, thus the assumption is made that Alice will not attempt to usurp Bob's identity. For instance, if Alice has D_1 and D_2 for associated devices and Bob has D_2 and D_3, they can both access the system with D_2 and their respective additional device. However, the authentication cannot be successfully performed by using only D_1 and D_3.

- Device duplication: A device can be purposefully duplicated to serve as a backup. One disadvantage is an additional attack vector for each duplicated device.

- Continuous Authentication: In the case where no interaction is required from the user, continuous authentication can have benefits. Indeed, session keys could have a shortened duration and the protocol could be repeated in a relatively short amount of time. Requiring a user to enter their password every minute or even every hour would incur a greater annoyance, thus would negatively impact the usability of systems with such a policy. Thus the importance of a zero-interaction implementation for continuous authentication is necessary for usability. As previously explained coauthentication can be a zero-interaction technique which would be suitable for continuous authentication.

- Multi-factor Authentication: An additional factor could be required to strengthen the authentication implementation. For example, a user could be required to scan their fingerprint in addition to succesfully completing the coauthentication process. Due to the increasing number of smartphones that support fingerprint scanners, such a scheme is possible, and as seen in Chapter 4, the usability may be better than coupling coauthentication with a password.

- Group authentication: A set of registered devices could be considered a registered group. This group could comprise multiple users required to authenticate with a few number of devices. For example, Alice and Bob each have two devices that could belong to the group and when the coauthentication is successfully performed, the authenticator identifies the user as a member of the group and not as an individual.

- Anonymous authentication: The notion of group authentication and shared devices can enable the anonymity of the individual users authenticated. Indeed, in the case where Alice and Bob share registered devices, the authenticator may not be able to identify a specific user of a group (or choose to not retain this information).

5.3.2 Authentication Methods Usability

The framework introduced here compiles compelling results in adopting such a framework for usability studies of authentication methods. Further investigation should strengthen this framework.

Investigating the usability of multi-factor authentication methods could generate unforeseen insight. Multi-factored authentication methods suffer from the disadvantage of each combined factor, but without running an experiment it is difficult to predict which combination would yield better result over another.

Several extensions exist for using the framework to evaluate authentication methods' usability in different contexts: through varying user activities, game design principles, or the authentication methods themselves.

Many user activities are obstructed by authentication requirements everyday, thus ensuing studies could modify the difficulty and the type of the activity simulated.

The activity simulated in the study presented was a conversation; however, simulating a conversation by having participants repeat words may not require enough cognitive load. The difficulty of this task can be adjusted to collect data and investigate the resulting effect(s). The following are potential modifications that would result in a different difficulty level:

1. The sets of words can be categorized based on a difficulty level.
2. The speed of the audio can be accelerated to increase the difficulty.
3. The sets of words during the experiment can appear in a randomized order.
4. The auditory task can be replaced with full sentences or questions.

All of these combinations are avenues to explore, to determine the effects of Dual-Task Interference and cognitive load while authenticating.

Additional game-design concepts can be included in the framework to further engage participant interest. For example, a leaderboard displayed during each break can provide feedback to participants on their performance compared to each other; several participants, during the trials, expressed interest in knowing how their scores compared to previous subjects. This particular gamification design is, therefore, one that should be considered in future related studies.

Another possible extension relates to setting a password for participants. In the experiment presented here, the participants' password was given, which was meant to prevent any "weak" password creations [85]. However, it cannot yet be determined whether having participants create their own passwords would significantly impact the study's results.

References

[1] Jay Ligatti, Cagri Cetin, Shamaria Engram, Jean-Baptiste Subils, and Dmitry Goldgof. Coauthentication. In *Proceedings of the 34rd Annual ACM Symposium on Applied Computing*, pages 1906–1915. ACM, 2019.

[2] Hassan Khan, Urs Hengartner, and Daniel Vogel. Usability and security perceptions of implicit authentication: Convenient, secure, sometimes annoying. In *SOUPS*, pages 225–239, 2015.

[3] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*, pages 553–567. IEEE, 2012.

[4] Lawrence O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, December 2003.

[5] Catherine S Weir, Gary Douglas, Martin Carruthers, and Mervyn Jack. User perceptions of security, convenience and usability for ebanking authentication tokens. *Computers & Security*, 28(1):47–62, 2009.

[6] Cagri Cetin, Jay Ligatti, and Dmitry Goldgof. SQL-Identifier injection attacks. In *2019 IEEE Conference on Communications and Network Security (CNS) (IEEE CNS 2019)*, 2019.

[7] Mehmet Aktukmak, Yasin Yilmaz, and Ismail Uysal. Quick and accurate attack detection in recommender systems through user attributes. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 348–352, New York, NY, USA, 2019. ACM.

[8] National Institute of Standards and Technology. Back to basics: Multi-factor authentication (MFA), November 2016. https://www.nist.gov/itl/tig/back-basics-multi-factor-authentication.

[9] Paul Grassi, James Fenton, Elaine Newton, Ray Perlner, Andrew Regenscheid, William Burr, Justin Richer, Naomi Lefkovitz, Jamie Danker, Yee-Yin Choong, Kristen Greene, and Mary Theofanos. NIST special publication 800-63B digital authentication guideline, June 2017. https://doi.org/10.6028/NIST.SP.800-63b.

[10] David M'Raihi, Salah Machani, Mingliang Pei, and Johan Rydell. TOTP: Time-based one-time password algorithm. RFC 6238, May 2011. http://www.rfc-editor.org/rfc/rfc6238.txt.

[11] M Angela Sasse, Sacha Brostoff, and Dirk Weirich. Transforming the 'weakest link'–a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, July 2001.

[12] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the International Conference on World Wide Web*, pages 657–666, May 2007.

[13] Philip G Inglesant and M Angela Sasse. The true cost of unusable password policies: Password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 383–392, April 2010.

[14] Florian Schaub, Ruben Deyhle, and Michael Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th international conference on mobile and ubiquitous multimedia*, page 10. ACM, 2012.

[15] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Usability and security of text passwords on mobile devices. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 527–539, May 2016.

[16] Nancy Gibbs. Your life is fully mobile. *TIME*, August 2012. http://techland.time.com/2012/08/16/your-life-is-fully-mobile/.

[17] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[18] Scott Ruoti and Kent E Seamons. Standard metrics and scenarios for usable authentication. In *WAY@ SOUPS*, 2016.

[19] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[20] Jean-Baptiste Subils, Joseph Perez, Peiwei Liu, Shamaria Engram, Cagri Cetin, Dmitry Goldgof, Natalie Ebner, Daniela Oliveira, and Jay Ligatti. A dual-task interference game-based experimental framework for comparing the usability of authentication methods. In *12th International Conference on Human System Interaction (HSI)*. IEEE, 2019.

[21] Harold Pashler. Dual-task interference and elementary mental mechanisms. *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, pages 245–264, 1993.

[22] Harold Pashler. Dual-task interference in simple tasks: data and theory. *Psychological bulletin*, 116(2):220, 1994.

[23] Juho Hamari. Do badges increase user activity? a field experiment on the effects of gamification. *Computers in human behavior*, 71:469–478, 2017.

[24] Gabriel Barata, Sandra Gama, Joaquim Jorge, and Daniel Gonçalves. Improving participation and learning with gamification. In *Proceedings of the First International Conference on gameful design, research, and applications*, pages 10–17. ACM, 2013.

[25] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences (HICSS)*, pages 3025–3034. IEEE, 2014.

[26] Bruno Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 82–96, June 2001.

[27] Bruno Blanchet. ProVerif: Cryptographic protocol verifier in the formal model, 2016. http://prosecco.gforge.inria.fr/personal/bblanche/proverif/.

[28] Kresimir Delac and Mislav Grgic. A survey of biometric recognition methods. In *46th International Symposium Electronics in Marine*, volume 46, pages 16–18, 2004.

[29] John Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[30] Jarred Adam Ligatti, Dmitry Goldgof, Cagri Cetin, and Jean-Baptiste Subils. System and methods for authentication using multiple devices, May 23 2017. US Patent 9,659,160.

[31] Jarred Adam Ligatti, Dmitry Goldgof, Cagri Cetin, and Jean-Baptiste Subils. Systems and methods for anonymous authentication using multiple devices, June 28 2016. US Patent 9,380,058.

[32] Jay Ligatti, Cagri Cetin, Shamaria Engram, Jean-Baptiste Subils, and Dmitry Goldgof. Coauthentication. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, April 2019.

[33] Jay Ligatti, Cagri Cetin, Shamaria Engram, Jean-Baptiste Subils, and Dmitry Goldgof. Coauthentication. Technical Report Technical Report CSE-SEC-092418, University of South Florida, Department of Computer Science and Engineering, September 2018. http://www.cse.usf.edu/~ligatti/papers/CoauthTR-092418.pdf.

[34] Jay Ligatti, Cagri Cetin, Shamaria Engram, Jean-Baptiste Subils, and Dmitry Goldgof. Coauthentication. Technical Report Auth-7-17-17, University of South Florida, Department of Computer Science and Engineering, July 2017. http://www.cse.usf.edu/~ligatti/papers/coauth-TR.pdf.

[35] Cagri Cetin. *Authentication and SQL-Injection Prevention Techniques in Web Applications*. PhD thesis, University of South Florida, 2019.

[36] Mark D Corner and Brian D Noble. Zero-Interaction authentication. In *Proceedings of the ACM International Conference on Mobile Computing and Networking*, pages 1–11, September 2002.

[37] Geneva Belford, Steve Bunch, John Day, Peter Alsberg, Deborah Brown, Enrique Grapa, David Healy, and John Mullen. A state-of-the-art report on network data management and related technology. Technical Report 150, Center for Advanced Computation, University of Illinois at Urbana-Champaign, April 1975. Page 132. https://archive.org/details/stateoftheartrep150belf.

[38] International Standards Organization. Information technology – Trusted platform module library – Part 1: Architecture. Technical report, August 2015. ISO/IEC 11889-1:2015. https://www.iso.org/standard/66510.html.

[39] Cagri Cetin and Jay Ligatti. ProVerif coauthentication files, December 2018. https://github.com/Coauthentication/FormalModels.

[40] Bruno Blanchet, Ben Smyth, Vincent Cheval, and Marc Sylvestre. ProVerif 1.98pl1: Automatic cryptographic protocol verifier, user manual and tutorial, December 2017. http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.

[41] Thomas Y.C. Woo and Simon S. Lam. A semantic model for authentication protocols. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 178–194, 1993.

[42] John Dunning. Taming the blue beast: A survey of bluetooth-based threats. *IEEE Security & Privacy*, 8(2):20–27, 2010.

[43] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Seyoung Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Can long passwords be secure and usable? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2927–2936. ACM, 2014.

[44] Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Alain Forget, Saranga Komanduri, Michelle L Mazurek, William Melicher, Sean M Segreti, and Blase Ur. A spoonful of sugar?: The impact of guidance and feedback on password-creation behavior. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2903–2912, April 2015.

[45] Department of Defense. *Nuclear Weapon Accident Response Procedures (NARP)*, September 1990. DoD 5100.52-M. https://fas.org/nuke/guide/usa/doctrine/dod/5100-52m/chap15.pdf.

[46] Margaret Woodward. Air force instruction 91-104, April 2013. https://fas.org/irp/doddir/usaf/afi91-104.pdf.

[47] Michael Reed, Paul Syverson, and David Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications*, 16(4):482–494, 1998.

[48] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[49] Mihir Bellare and Gregory Neven. Identity-based multi-signatures from RSA. In *Proceedings of the Cryptographers' Track at the RSA Conference*, pages 145–162. Springer, February 2007.

[50] Radhesh Krishnan Konoth, Victor van der Veen, and Herbert Bos. How anywhere computing just killed your phone-based two-factor authentication, 2016. http://fc16.ifca.ai/preproceedings/24_Konoth.pdf.

[51] Charles McColgan. Issues with SMS deprecation rationale, September 2016. https://github.com/usnistgov/800-63-3/issues/351.

[52] Tiano Freixas Lopez Lecube, Josh Miller, Thomas Jaeger, Sam Oh, Wenhan Zhao, and Eric Min. Multi-device authentication, 2015. US Patent Application 2017/0093846 A1.

[53] Joseph Bonneau, Cormac Herley, Paul C van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE Symposium on Security and Privacy*, pages 553–567, 2012.

[54] Daniel Peralta, Mikel Galar, Isaac Triguero, Daniel Paternain, Salvador García, Edurne Barrenechea, José M Benítez, Humberto Bustince, and Francisco Herrera. A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation. *Information Sciences*, 315:67–87, 2015.

[55] Mitja Rutnik. Over 70 percent of smartphones shipped in 2018 will have fingerprint sensors — report, 2017.

[56] Yulong Zhang, Zhaonfeng Chen, Hui Xue, and Tao Wei. Fingerprints on mobile devices: Abusing and leaking. In *Black Hat Conference*, 2015.

[57] Android developers, fingerprint specifications. https://source.android.com/security/authentication/fingerprint-hal.

[58] Burt Kaliski. Pkcs# 5: Password-based cryptography specification version 2.0. Technical report, 2000.

[59] Kathleen Moriarty, Burt Kaliski, and Andreas Rusch. Pkcs# 5: Password-based cryptography specification version 2.1. Technical report, 2017.

[60] Meltem Sönmez Turan, Elaine Barker, William Burr, and Lily Chen. Recommendation for password-based key derivation. *NIST special publication*, 800:132, 2010.

[61] Wayne C Summers and Edward Bosworth. Password policy: the good, the bad, and the ugly. In *Proceedings of the winter international synposium on Information and communication technologies*, pages 1–6. Trinity College Dublin, 2004.

[62] Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 2. ACM, 2010.

[63] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 345–352. ACM, 2000.

[64] Kasper Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies*, 64(2):79–102, 2006.

[65] Nigel Bevan. Measuring usability as quality of use. *Software Quality Journal*, 4(2):115–130, 1995.

[66] Kevin A Juang, Sanjay Ranganayakulu, and Joel S Greenstein. Using system-generated mnemonics to improve the usability and security of password authentication. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, pages 506–510. SAGE Publications Sage CA: Los Angeles, CA, 2012.

[67] Anne M Gadermann, Martin Guhn, and Bruno D Zumbo. Estimating ordinal reliability for likert-type and ordinal item response data: A conceptual, empirical, and practical guide. *Practical Assessment, Research & Evaluation*, 17(3):1–13, 2012.

[68] Thomas S Tullis and Jacqueline N Stetson. A comparison of questionnaires for assessing website usability. In *Usability professional association conference*, volume 1, 2004.

[69] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy Van Der Horst, and Kent Seamons. Confused johnny: when automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, page 5. ACM, 2013.

[70] John P Chin, Virginia A Diehl, and Kent L Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–218. ACM, 1988.

[71] James R Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[72] Joey Benedek and Trish Miner. Measuring desirability: New methods for evaluating desirability in a usability lab setting. *Proceedings of Usability Professionals Association*, 2003(8-12):57, 2002.

[73] Markus Jakobsson, Elaine Shi, Philippe Golle, and Richard Chow. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX conference on Hot topics in security*, pages 9–9, 2009.

[74] John M Carroll and John C Thomas. Fun. *ACM SIGCHI Bulletin*, 19(3):21–24, 1988.

[75] Noam Tractinsky. Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 115–122. ACM, 1997.

[76] Jenny Preece. Sociability and usability in online communities: Determining and measuring success. *Behaviour & Information Technology*, 20(5):347–356, 2001.

[77] Donna L Hoffman and Thomas P Novak. Marketing in hypermedia computer-mediated environments: Conceptual foundations. *Journal of marketing*, 60(3):50–68, 1996.

[78] Judy Jeng. Usability assessment of academic digital libraries: effectiveness, efficiency, satisfaction, and learnability. *Libri*, 55(2-3):96–121, 2005.

[79] Christina Braz and Jean-Marc Robert. Security and usability: the case of the user authentication methods. In *Proceedings of the 18th Conference on l'Interaction Homme-Machine*, pages 199–203. ACM, 2006.

[80] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010.

[81] Sean M Segreti, William Melicher, Saranga Komanduri, Darya Melicher, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Diversify to survive: Making passwords stronger with adaptive policies. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[82] Hana Habib, Pardis Emami Naeini, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. User behaviors and attitudes under password expiration policies. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS)*, 2018.

[83] Yinqian Zhang, Fabian Monrose, and Michael K Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 176–186. ACM, 2010.

[84] Yee-Yin Choong, Mary Theofanos, and Hung-Kung Liu. *United States Federal Employees' Password Management Behaviors: A Department of Commerce Case Study*. US Department of Commerce, National Institute of Standards and Technology, 2014.

[85] Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Honey, i shrunk the keys: influences of mobile devices on password composition and authentication performance. In *Proceedings of the 8th nordic conference on human-computer interaction: fun, fast, foundational*, pages 461–470. ACM, 2014.

[86] Václav Matyáš and Zdeněk Říha. Biometric authentication—security and usability. In *Advanced Communications and Multimedia Security*, pages 227–239. Springer, 2002.

[87] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, 2011.

[88] Ravi Subban and Dattatreya P Mankame. A study of biometric approach using fingerprint recognition. *Lecture notes on software engineering*, 1(2):209, 2013.

[89] Mehmet Aktukmak, Yasin Yilmaz, and Ismail Uysal. A probabilistic framework to incorporate mixed-data type features: Matrix factorization with multimodal side information. *Neurocomputing*, 08 2019.

[90] M. Aktukmak, S. Mercier, and I. Uysal. A neural net framework for accumulative feature-based matrix completion. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, July 2018.

[91] S. Fang, I. Markwood, and Y. Liu. Wireless-assisted key establishment leveraging channel manipulation. *IEEE Transactions on Mobile Computing*, 2019.

[92] S. Fang, T. Wang, Y. Liu, S. Zhao, and Z. Lu. Entrapment for wireless eavesdroppers. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2530–2538, April 2019.

[93] S. Fang, I. Markwood, and Y. Liu. Manipulatable wireless key establishment. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, Oct 2017.

[94] T. Wang, Y. Liu, T. Hou, Q. Pei, and S. Fang. Signal entanglement based pinpoint waveforming for location-restricted service access control. *IEEE Transactions on Dependable and Secure Computing*, 15(5):853–867, Sep. 2018.

[95] S. Fang, Y. Liu, W. Shen, H. Zhu, and T. Wang. Virtual multipath attack and defense for location distinction in wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):566–580, Feb 2017.

[96] S. Fang, Y. Liu, and P. Ning. Mimicry attacks against wireless link signature and new defense using time-synched link signature. *IEEE Transactions on Information Forensics and Security*, 11(7):1515–1527, July 2016.

[97] S. Fang, Y. Liu, and P. Ning. Wireless communications under broadband reactive jamming attacks. *IEEE Transactions on Dependable and Secure Computing*, 13(3):394–408, May 2016.

[98] Song Fang, Ian Markwood, Yao Liu, Shangqing Zhao, Zhuo Lu, and Haojin Zhu. No training hurdles: Fast training-agnostic attacks to infer your typing. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’18, pages 1747–1760, Toronto, Canada, 2018. ACM.

[99] Song Fang, Yao Liu, Wenbo Shen, and Haojin Zhu. Where are you from?: Confusing location distinction using virtual multipath camouflage. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 225–236, Maui, Hawaii, USA, 2014. ACM.

Appendix A: Participant Information Questionnaire

- Demographics

1. Handedness *
   - ○ Left
   - ○ Right
   - ○ Ambidextrous
2. Gender *
   - ○ Male
   - ○ Female
   - ○ Prefer not to answer
3. Enter your age in number of years *

- Education

4. Native Language *
5. If not English, at what age approximately did you begin to speak/learn English
6. Are you bilingual *
   - ○ Yes
   - ○ No

7. Number of years of education *

8. What is your major? (If you are currently in college) *

- Eligibility

9. Do you have any hearing difficulty? *

   - Yes

   - No

10. If yes, please explain:

11. Do you have any visual difficulty? *

    - Yes

    - No

12. If yes, please explain

13. Do you wear glasses? *

    - Yes

    - No

    - Sometimes

- Ethnic Category

14. Please check the most appropriate category *

    - Hispanic or Latino

    - Not Hispanic or Latino

- Racial Category

15. Please select all that apply *

    □ American Indian or Alaskan Native

    □ Asian

    □ Black or African American

    □ Native Hawaiian or Other Pacific Islander

    □ White

- Technology and Authentication Experience

16. Approximately how many hours per day do you use a computer (desktop/laptop)? *

17. Approximately how many hours per day do you use a smartphone? *

18. Approximately how many hours per day do you use an electronic device other than the ones mentioned above? *

19. If your answer is greater than 0 please list which device(s)?

20. Approximately how many times per day do you type a ”long” password to log into a device or service? (i.e., at least 7 typed characters) *

21. Approximately how many times per day do you type a ”short” password to log into a device or service? (i.e., at most 6 typed characters, or passwords based on swiping a particular pattern, or making some sort of gesture. Including passwords and PINs to log into phones, ATM, etc.) *

22. Approximately how many times per day do you log into a device or service using your fingerprint? *

Appendix B: Authentication Experience Questionnaire

1. Could you hear the words properly during the auditory tasks?

   ○ Yes

   ○ No

2. I found the audio tasks got in the way of me being able to complete the authentication tasks

1	2	3	4	5
○	○	○	○	○

3. I think that I would like to use this system frequently

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

4. I found the authentication method unnecessarily complex

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

5. I thought the system was easy to use

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

6. I would need the support of a technical person to be able to use this authentication method

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

7. The various functions in this authentication method were well integrated

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

8. I thought there was too much inconsistency in this authentication method

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

9. Most people would learn to use this authentication method very quickly

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

10. This authentication method is very cumbersome to use

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

11. I felt very confident using this authentication method

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

12. I needed to learn a lot of things before I could get going with this authentication method

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

13. I could complete this authentication task even while I am distracted

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

14. This authentication method was easy to use

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

15. This authentication method was fast to use

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

16. This authentication method required attention/focus

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

17. This authentication method seemed secure

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

18. Please rank the authentication methods in order of your preference from 1 (most preferred) to 5 (least preferred)

	1	2	3	4	5
Username/password laptop	○	○	○	○	○
Coauthentication laptop	○	○	○	○	○
Username/password phone	○	○	○	○	○
Coauthentication phone	○	○	○	○	○
Fingerprint phone	○	○	○	○	○

Appendix C: Auditory Task Words

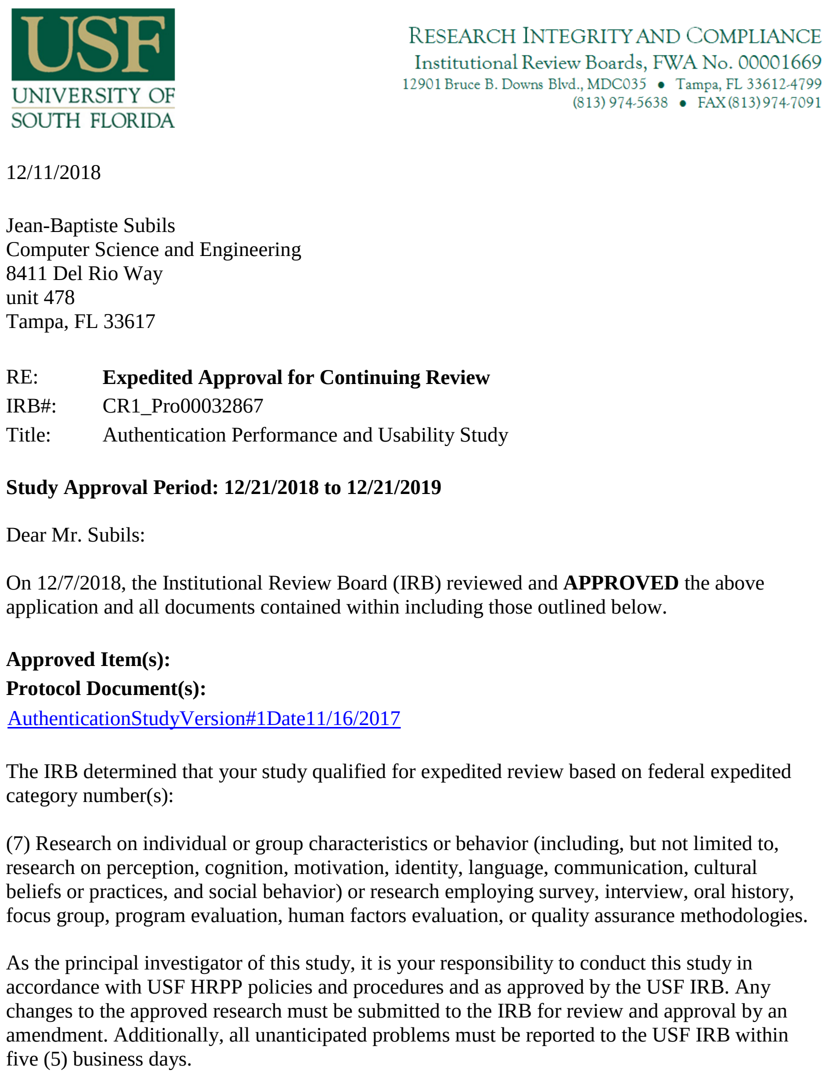
- Audio Practice: an, bake, in, duck, boat, knee, tray, face, sick

- Multi-tasking game practice: sale, must, pot, math, bad, pay, food, cane, hair, paste, doll, keep, led, heart, miss, tree, pop, cup, nose, sink, slip, fog, cart, deck, buzz, sheep, loud, hurt, pass, bee, drop, quick, nest, thank, sled, frog, park, neck, bus

- Multi-tasking game conversation: stop, tea, you, draw, need, oil, book, tick, goat, we, stay, base, fit, well, best, lot, pat, bug, may, seat, pain, fast, wall, ship, load, dirt, grass, see, top, quit, pest, tank, slide, grade, mark, poke, fuss, spot, toe, rat, show, lead, fell, coat, glove, road, hope, lake, tell, wet, feel, bad, care, fruit, nest, chip, gave, five, ice, ran, frog, soft, pink, tent, milk, bake, room, ant, woke, hand, sheep, read, bed, bus, mud, night, some, gone, move, gold, mean, last, hat, meet, on, weed, due, am, say, mind, off, has, mile, must, bat, two, row, shop, lie, when, wade, hose, grow, are, south, sew, date, touch, hop, wipe, truck, dark, day, pot, save, right, does, thing, cry, park, neck, key, fat, shoe, tall, feed

Appendix D: Institutional Review Board Authorization

The IRB approval below is for the usability experiment.



USF
UNIVERSITY OF
SOUTH FLORIDA

RESEARCH INTEGRITY AND COMPLIANCE
Institutional Review Boards, FWA No. 00001669
12901 Bruce B. Downs Blvd., MDC035 • Tampa, FL 33612-4799
(813) 974-5638 • FAX (813) 974-7091

12/11/2018

Jean-Baptiste Subils
Computer Science and Engineering
8411 Del Rio Way
unit 478
Tampa, FL 33617

RE: **Expedited Approval for Continuing Review**
IRB#: CR1_Pro00032867
Title: Authentication Performance and Usability Study

Study Approval Period: 12/21/2018 to 12/21/2019

Dear Mr. Subils:

On 12/7/2018, the Institutional Review Board (IRB) reviewed and **APPROVED** the above application and all documents contained within including those outlined below.

Approved Item(s):
Protocol Document(s):
AuthenticationStudyVersion#1Date11/16/2017

The IRB determined that your study qualified for expedited review based on federal expedited category number(s):

(7) Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and social behavior) or research employing survey, interview, oral history, focus group, program evaluation, human factors evaluation, or quality assurance methodologies.

As the principal investigator of this study, it is your responsibility to conduct this study in accordance with USF HRPP policies and procedures and as approved by the USF IRB. Any changes to the approved research must be submitted to the IRB for review and approval by an amendment. Additionally, all unanticipated problems must be reported to the USF IRB within five (5) business days.

We appreciate your dedication to the ethical conduct of human subject research at the University of South Florida and your continued commitment to human research protections. If you have any questions regarding this matter, please call 813-974-5638.

Sincerely,

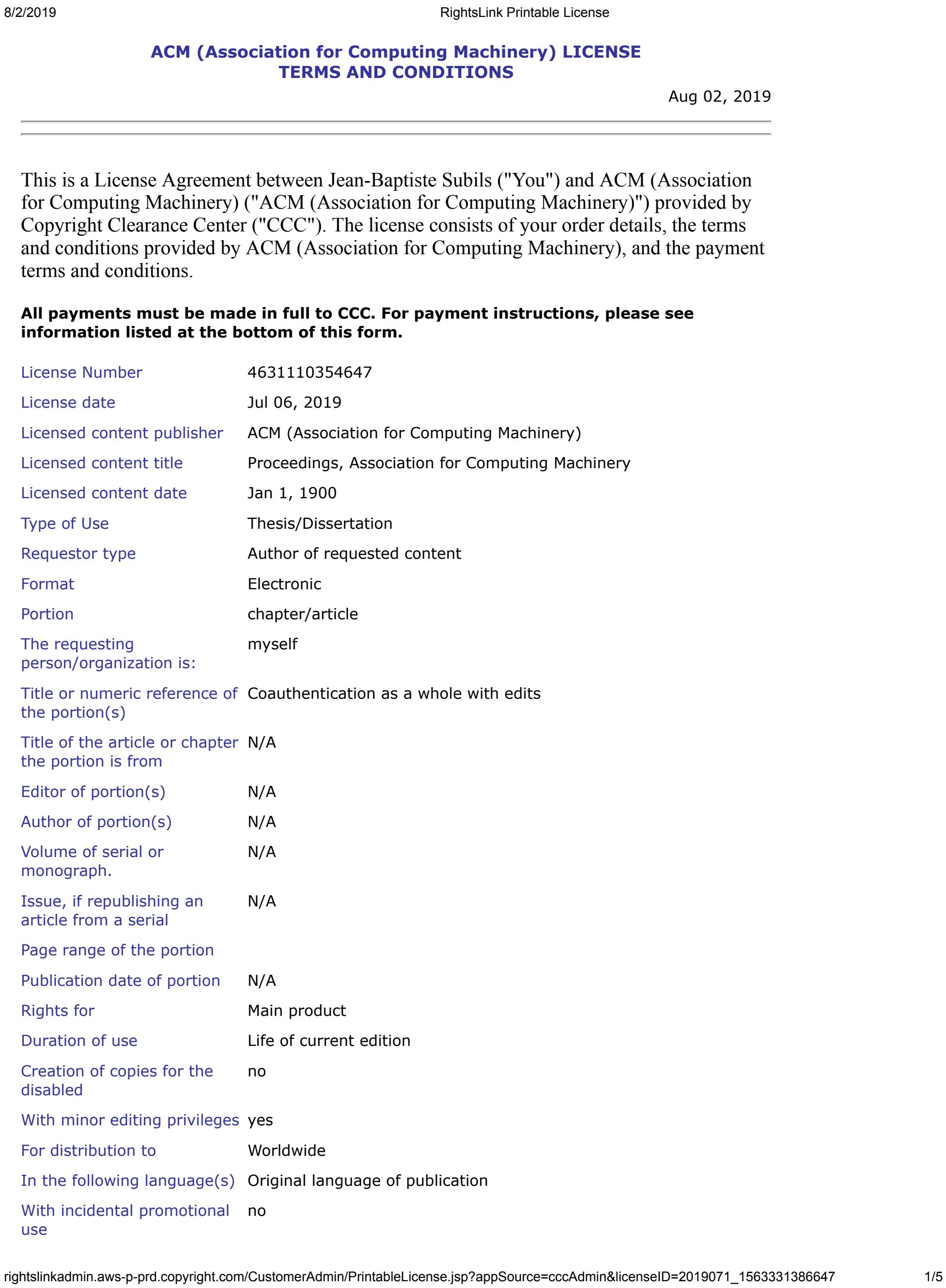
[redacted]

Kristen Salomon, Ph.D., Chairperson
USF Institutional Review Board

Appendix E: Copyright Permissions

The chapters 3 and 4, sections 1.3, 1.4.2 and 5.3.2, and their figures and tables are, in part or as a whole, copyrighted material © 2019 IEEE [20].In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of South Florida's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

The permission below is for the Coauthentication published paper [32] which is used in parts for Chapter 5 and its figures and tables, Sections 1.1, 5.2, and 5.3.1.



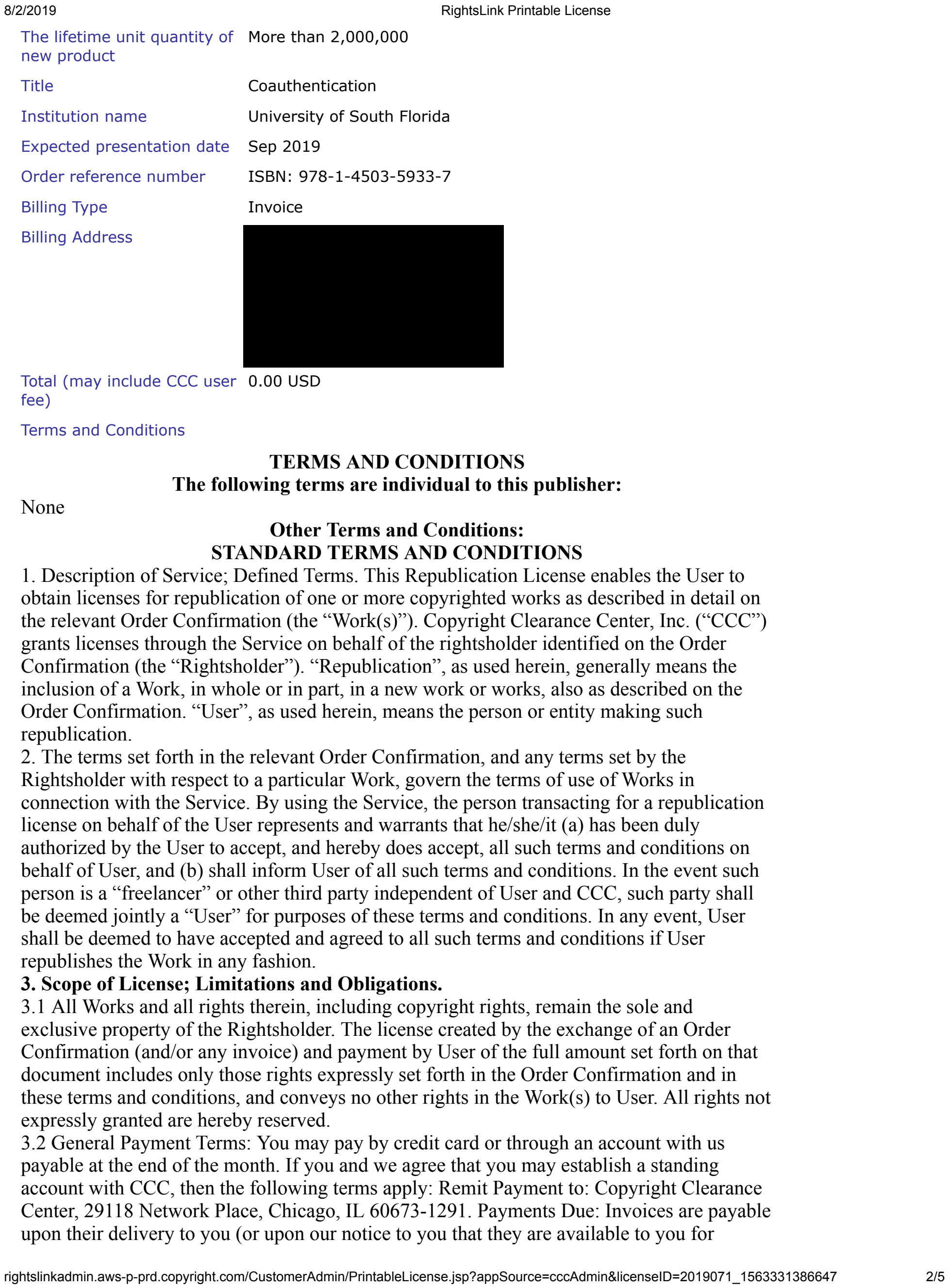
ACM (Association for Computing Machinery) LICENSE TERMS AND CONDITIONS

Aug 02, 2019

This is a License Agreement between Jean-Baptiste Subils ("You") and ACM (Association for Computing Machinery) ("ACM (Association for Computing Machinery)") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by ACM (Association for Computing Machinery), and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	4631110354647
License date	Jul 06, 2019
Licensed content publisher	ACM (Association for Computing Machinery)
Licensed content title	Proceedings, Association for Computing Machinery
Licensed content date	Jan 1, 1900
Type of Use	Thesis/Dissertation
Requestor type	Author of requested content
Format	Electronic
Portion	chapter/article
The requesting person/organization is:	myself
Title or numeric reference of the portion(s)	Coauthentication as a whole with edits
Title of the article or chapter the portion is from	N/A
Editor of portion(s)	N/A
Author of portion(s)	N/A
Volume of serial or monograph.	N/A
Issue, if republishing an article from a serial	N/A
Page range of the portion	
Publication date of portion	N/A
Rights for	Main product
Duration of use	Life of current edition
Creation of copies for the disabled	no
With minor editing privileges	yes
For distribution to	Worldwide
In the following language(s)	Original language of publication
With incidental promotional use	no



The lifetime unit quantity of new product	More than 2,000,000
Title	Coauthentication
Institution name	University of South Florida
Expected presentation date	Sep 2019
Order reference number	ISBN: 978-1-4503-5933-7
Billing Type	Invoice
Billing Address	[redacted]
Total (may include CCC user fee)	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS
The following terms are individual to this publisher:

None

Other Terms and Conditions:
STANDARD TERMS AND CONDITIONS

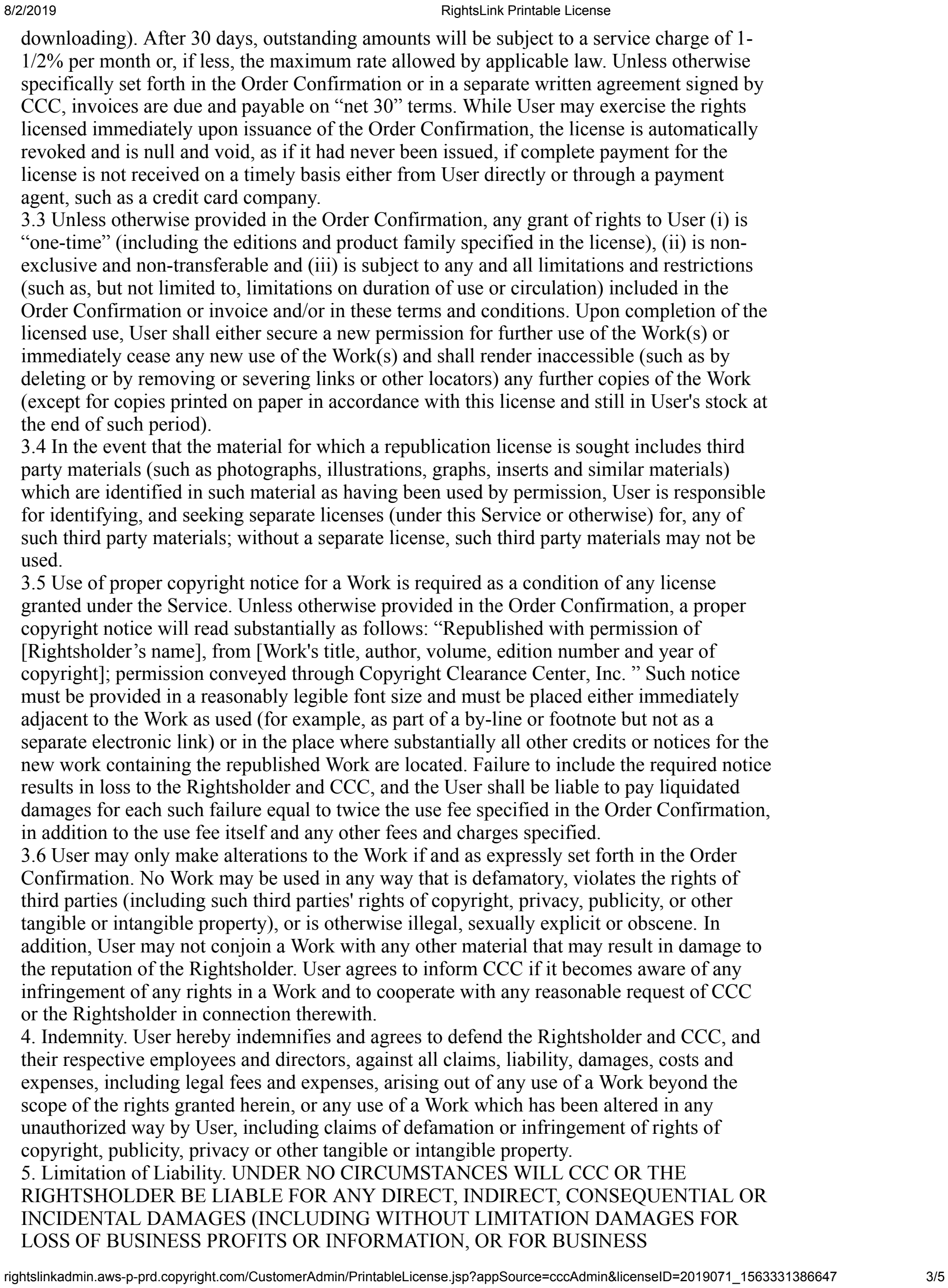
1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the “Work(s)”). Copyright Clearance Center, Inc. (“CCC”) grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the “Rightsholder”). “Republication”, as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. “User”, as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a “freelancer” or other third party independent of User and CCC, such party shall be deemed jointly a “User” for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

3.1 All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2 General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for



downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on “net 30” terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3 Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is “one-time” (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

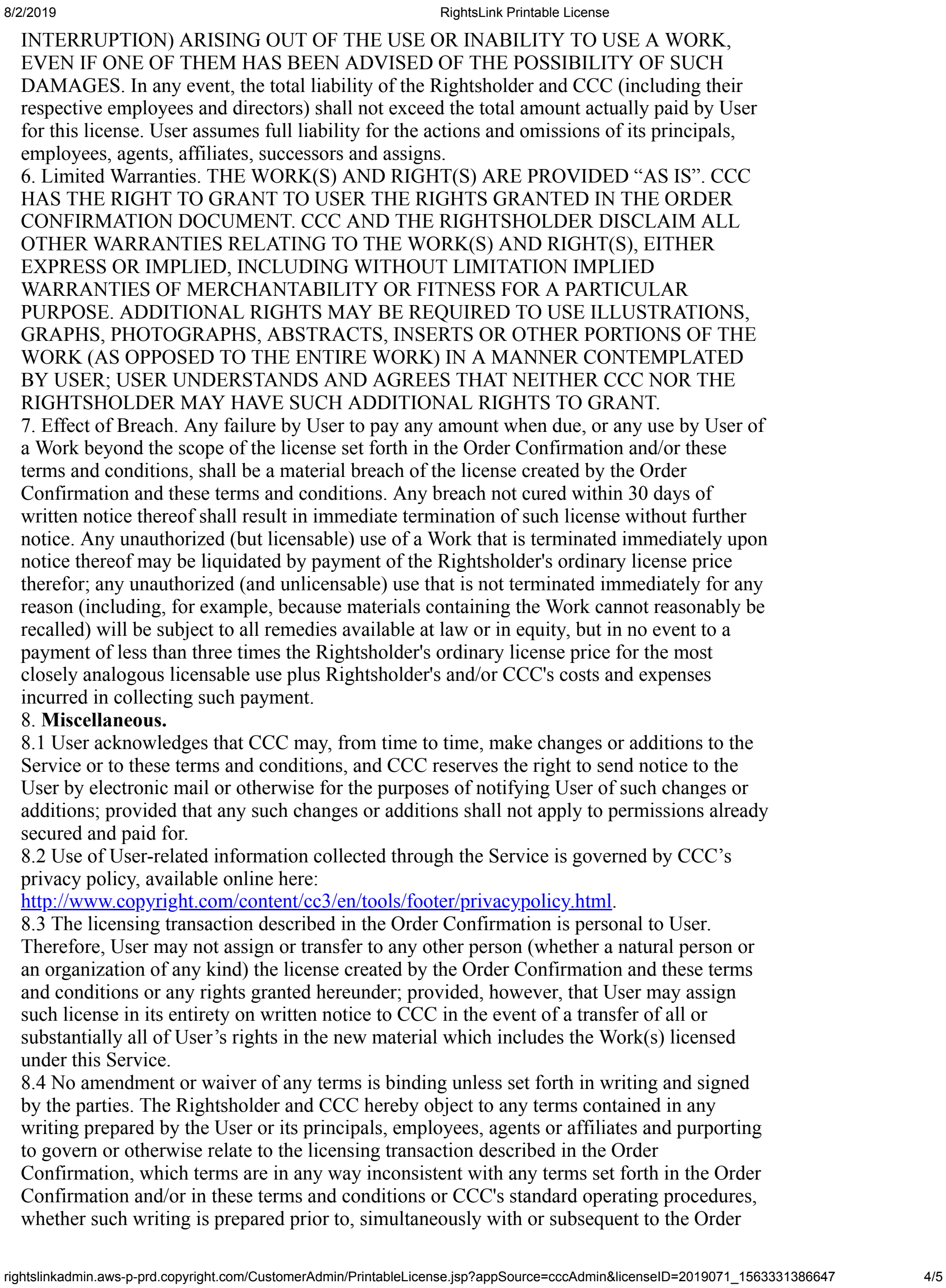
3.4 In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5 Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: “Republished with permission of [Rightsholder’s name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. ” Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6 User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS



INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED “AS IS”. CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

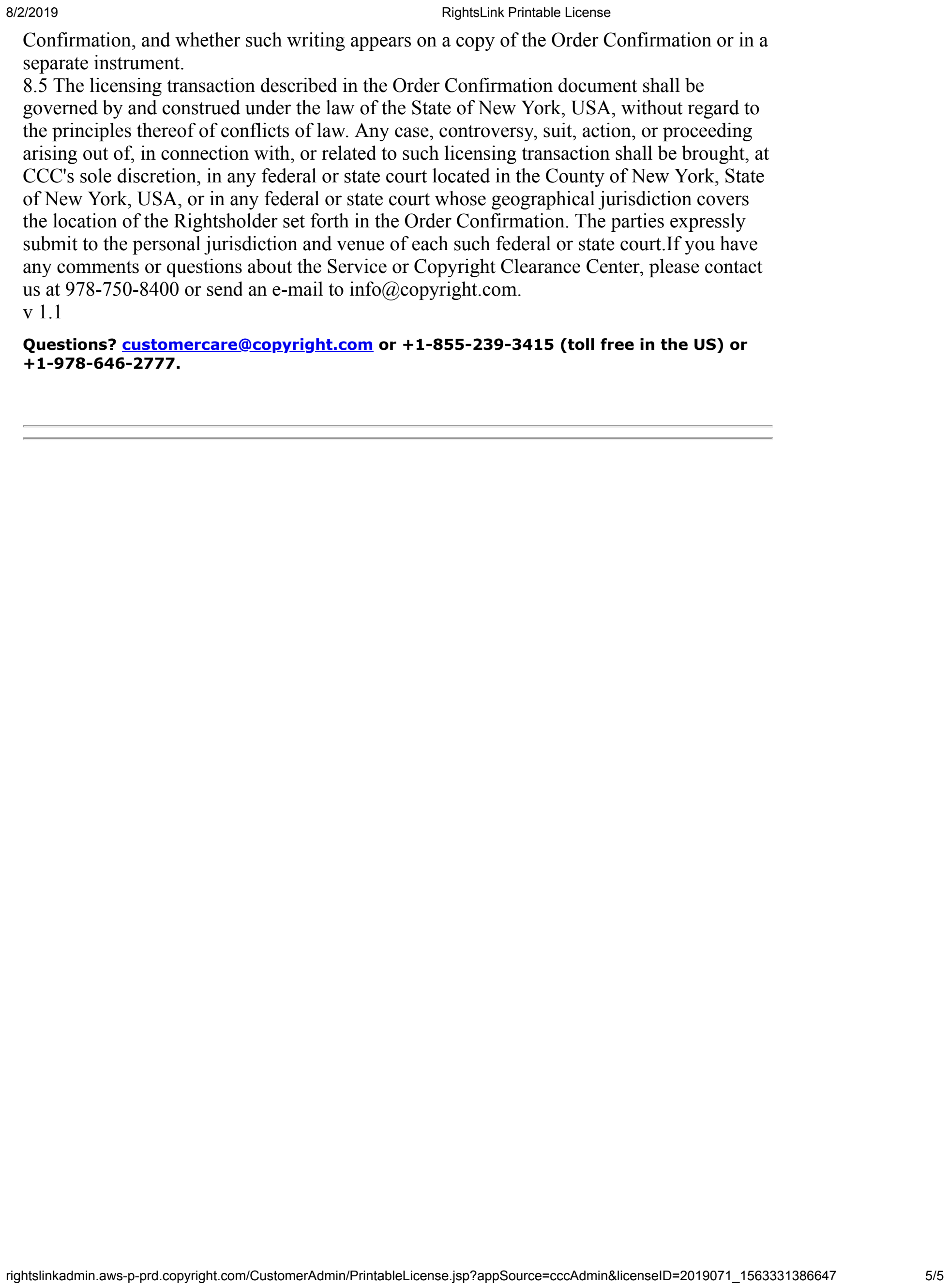
8. **Miscellaneous.**

8.1 User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

8.2 Use of User-related information collected through the Service is governed by CCC’s privacy policy, available online here: http://www.copyright.com/content/cc3/en/tools/footer/privacypolicy.html.

8.3 The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User’s rights in the new material which includes the Work(s) licensed under this Service.

8.4 No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order



Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

8.5 The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to info@copyright.com.

v 1.1

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.