Acknowledgments

The authors gratefully acknowledge Kaitlin Boeckl for her artistic graphics contributions to all volumes in the SP 800-63 suite and the contributions of our many reviewers, including Joni Brennan from the Digital ID & Authentication Council of Canada (DIACC), Kat Megas, Ellen Nadeau, and Ben Piccarreta from NIST, and Ryan Galluzzo and Danna Gabel O'Rourke from Deloitte & Touche LLP.

The authors would also like to acknowledge the thought leadership and innovation of the original authors: Donna F. Dodson, W. Timothy Polk, Sarbari Gupta, and Emad A. Nabbus. Without their tireless efforts, we would not have had the incredible baseline from which to evolve 800-63 to the document it is today. In addition, special thanks to the Federal Privacy Council's Digital Authentication Task Force for the contributions to the development of privacy requirements and considerations.

Requirements Notation and Conventions

The terms "SHALL" and "SHALL NOT" indicate requirements to be followed strictly in order to conform to the publication and from which no deviation is permitted.

The terms "SHOULD" and "SHOULD NOT" indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited.

The terms "MAY" and "NEED NOT" indicate a course of action permissible within the limits of the publication.

The terms "CAN" and "CANNOT" indicate a possibility or capability, whether material, physical or causal or, in the negative, the absence of that possibility or capability.

Appendix A—Strength of Memorized Secrets

This appendix is informative.

Throughout this appendix, the word "password" is used for ease of discussion. Where used, it should be interpreted to include passphrases and PINs as well as passwords.

A.1 Introduction

Despite widespread frustration with the use of passwords from both a usability and security standpoint, they remain a very widely used form of authentication [Persistence]. Humans, however, have only a limited ability to memorize complex, arbitrary secrets, so they often choose passwords that can be easily guessed. To address the resultant security concerns, online services have introduced rules in an effort to increase the complexity of these memorized secrets. The most notable form of these is composition rules, which require the user to choose passwords constructed using a mix of character types, such as at least one digit, uppercase letter, and symbol. However, analyses of breached password databases reveal that the benefit of such rules is not nearly as significant as initially thought [Policies], although the impact on usability and memorability is severe.

Complexity of user-chosen passwords has often been characterized using the information theory concept of entropy [Shannon]. While entropy can be readily calculated for data having deterministic distribution functions, estimating the entropy for user-chosen passwords is difficult and past efforts to do so have not been particularly accurate. For this reason, a different and somewhat simpler approach, based primarily on password length, is presented herein.

Many attacks associated with the use of passwords are not affected by password complexity and length. Keystroke logging, phishing, and social engineering attacks are equally effective on lengthy, complex passwords as simple ones. These attacks are outside the scope of this Appendix.

A.2 Length

Password length has been found to be a primary factor in characterizing password strength [Strength] [Composition]. Passwords that are too short yield to brute force attacks as well as to dictionary attacks using words and commonly chosen passwords.

The minimum password length that should be required depends to a large extent on the threat model being addressed. Online attacks where the attacker attempts to log in by guessing the password can be mitigated by limiting the rate of login attempts permitted. In order to prevent an attacker (or a persistent claimant with poor typing skills) from easily inflicting a denial-of-service attack on the subscriber by making many incorrect guesses, passwords need to be complex enough that rate limiting does not occur after a modest number of erroneous attempts, but does occur before there is a significant chance of a successful guess.

Offline attacks are sometimes possible when one or more hashed passwords is obtained by the attacker through a database breach. The ability of the attacker to determine one or more users' passwords depends on the way in which the password is stored. Commonly, passwords are salted

with a random value and hashed, preferably using a computationally expensive algorithm. Even with such measures, the current ability of attackers to compute many billions of hashes per second with no rate limiting requires passwords intended to resist such attacks to be orders of magnitude more complex than those that are expected to resist only online attacks.

Users should be encouraged to make their passwords as lengthy as they want, within reason. Since the size of a hashed password is independent of its length, there is no reason not to permit the use of lengthy passwords (or pass phrases) if the user wishes. Extremely long passwords (perhaps megabytes in length) could conceivably require excessive processing time to hash, so it is reasonable to have some limit.

A.3 Complexity

As noted above, composition rules are commonly used in an attempt to increase the difficulty of guessing user-chosen passwords. Research has shown, however, that users respond in very predictable ways to the requirements imposed by composition rules [Policies]. For example, a user that might have chosen "password" as their password would be relatively likely to choose "Password1" if required to include an uppercase letter and a number, or "Password1!" if a symbol is also required.

Users also express frustration when attempts to create complex passwords are rejected by online services. Many services reject passwords with spaces and various special characters. In some cases, the special characters that are not accepted might be an effort to avoid attacks like SQL injection that depend on those characters. But a properly hashed password would not be sent intact to a database in any case, so such precautions are unnecessary. Users should also be able to include space characters to allow the use of phrases. Spaces themselves, however, add little to the complexity of passwords and may introduce usability issues (e.g., the undetected use of two spaces rather than one), so it may be beneficial to remove repeated spaces in typed passwords prior to verification.

Users' password choices are very predictable, so attackers are likely to guess passwords that have been successful in the past. These include dictionary words and passwords from previous breaches, such as the "Password1!" example above. For this reason, it is recommended that passwords chosen by users be compared against a "black list" of unacceptable passwords. This list should include passwords from previous breach corpuses, dictionary words, and specific words (such as the name of the service itself) that users are likely to choose. Since user choice of passwords will also be governed by a minimum length requirement, this dictionary need only include entries meeting that requirement.

Highly complex memorized secrets introduce a new potential vulnerability: they are less likely to be memorable, and it is more likely that they will be written down or stored electronically in an unsafe manner. While these practices are not necessarily vulnerable, statistically some methods of recording such secrets will be. This is an additional motivation not to require excessively long or complex memorized secrets.

A.4 Randomly-Chosen Secrets

Another factor that determines the strength of memorized secrets is the process by which they are generated. Secrets that are randomly chosen (in most cases by the verifier or CSP) and are uniformly distributed will be more difficult to guess or brute-force attack than user-chosen secrets meeting the same length and complexity requirements. Accordingly, at LOA2, SP 800-63-2 permitted the use of randomly generated PINs with 6 or more digits while requiring user-chosen memorized secrets to be a minimum of 8 characters long.

As discussed above, the threat model being addressed with memorized secret length requirements includes rate-limited online attacks, but not offline attacks. With this limitation, 6 digit randomly-generated PINs are still considered adequate for memorized secrets.

A.5 Summary

Length and complexity requirements beyond those recommended here significantly increase the difficulty of memorized secrets and increase user frustration. As a result, users often work around these restrictions in a way that is counterproductive. Furthermore, other mitigations such as blacklists, secure hashed storage, and rate limiting are more effective at preventing modern brute-force attacks. Therefore, no additional complexity requirements are imposed.

5 Authenticator and Verifier Requirements

This section is normative.

This section provides the detailed requirements specific to each type of authenticator. With the exception of reauthentication requirements specified in Section 4 and the requirement for verifier impersonation resistance at AAL3 described in Section 5.2.5, the technical requirements for each of the authenticator types are the same regardless of the AAL at which the authenticator is used.

5.1 Requirements by Authenticator Type

5.1.1 Memorized Secrets

A Memorized Secret authenticator — commonly referred to as a *password* or, if numeric, a *PIN* — is a secret value intended to be chosen and memorized by the user. Memorized secrets need to be of sufficient complexity and secrecy that it would be impractical for an attacker to guess or otherwise discover the correct secret value. A memorized secret is *something you know*.

5.1.1.1 Memorized Secret Authenticators

Memorized secrets SHALL be at least 8 characters in length if chosen by the subscriber. Memorized secrets chosen randomly by the CSP or verifier SHALL be at least 6 characters in length and MAY be entirely numeric. If the CSP or verifier disallows a chosen memorized secret based on its appearance on a blacklist of compromised values, the subscriber SHALL be required to choose a different memorized secret. No other complexity requirements for memorized secrets SHOULD be imposed. A rationale for this is presented in Appendix A *Strength of Memorized Secrets*.

5.1.1.2 Memorized Secret Verifiers

Verifiers SHALL require subscriber-chosen memorized secrets to be at least 8 characters in length. Verifiers SHOULD permit subscriber-chosen memorized secrets at least 64 characters in length. All printing ASCII [RFC 20] characters as well as the space character SHOULD be acceptable in memorized secrets. Unicode [ISO/ISC 10646] characters SHOULD be accepted as well. To make allowances for likely mistyping, verifiers MAY replace multiple consecutive space characters with a single space character prior to verification, provided that the result is at least 8 characters in length. Truncation of the secret SHALL NOT be performed. For purposes of the above length requirements, each Unicode code point SHALL be counted as a single character.

If Unicode characters are accepted in memorized secrets, the verifier SHOULD apply the Normalization Process for Stabilized Strings using either the NFKC or NFKD normalization defined in Section 12.1 of Unicode Standard Annex 15 [UAX 15]. This process is applied before hashing the byte string representing the memorized secret. Subscribers choosing memorized secrets containing Unicode characters SHOULD be advised that some characters may be represented differently by some endpoints, which can affect their ability to authenticate successfully.

Memorized secrets that are randomly chosen by the CSP (e.g., at enrollment) or by the verifier (e.g., when a user requests a new PIN) SHALL be at least 6 characters in length and SHALL be generated using an approved random bit generator [SP 800-90Ar1].

Memorized secret verifiers SHALL NOT permit the subscriber to store a "hint" that is accessible to an unauthenticated claimant. Verifiers SHALL NOT prompt subscribers to use specific types of information (e.g., "What was the name of your first pet?") when choosing memorized secrets.

When processing requests to establish and change memorized secrets, verifiers SHALL compare the prospective secrets against a list that contains values known to be commonly-used, expected, or compromised. For example, the list MAY include, but is not limited to:

- Passwords obtained from previous breach corpuses.
- Dictionary words.
- Repetitive or sequential characters (e.g. 'aaaaaa', '1234abcd').
- Context-specific words, such as the name of the service, the username, and derivatives thereof.

If the chosen secret is found in the list, the CSP or verifier SHALL advise the subscriber that they need to select a different secret, SHALL provide the reason for rejection, and SHALL require the subscriber to choose a different value.

Verifiers SHOULD offer guidance to the subscriber, such as a password-strength meter [Meters], to assist the user in choosing a strong memorized secret. This is particularly important following the rejection of a memorized secret on the above list as it discourages trivial modification of listed (and likely very weak) memorized secrets [Blacklists].

Verifiers SHALL implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber's account as described in Section 5.2.2.

Verifiers SHOULD NOT impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets. Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator.

Verifiers SHOULD permit claimants to use "paste" functionality when entering a memorized secret. This facilitates the use of password managers, which are widely used and in many cases increase the likelihood that users will choose stronger memorized secrets.

In order to assist the claimant in successfully entering a memorized secret, the verifier SHOULD offer an option to display the secret — rather than a series of dots or asterisks — until it is entered. This allows the claimant to verify their entry if they are in a location where their screen is unlikely to be observed. The verifier MAY also permit the user's device to display individual entered characters for a short time after each character is typed to verify correct entry. This is particularly applicable on mobile devices.

The verifier SHALL use approved encryption and an authenticated protected channel when requesting memorized secrets in order to provide resistance to eavesdropping and MitM attacks.

Verifiers SHALL store memorized secrets in a form that is resistant to offline attacks. Memorized secrets SHALL be salted and hashed using a suitable one-way key derivation function. Key derivation functions take a password, a salt, and a cost factor as inputs then generate a password hash. Their purpose is to make each password guessing trial by an attacker who has obtained a password hash file expensive and therefore the cost of a guessing attack high or prohibitive. Examples of suitable key derivation functions include Password-based Key Derivation Function 2 (PBKDF2) [SP 800-132] and Balloon [BALLOON]. A memory-hard function SHOULD be used because it increases the cost of an attack. The key derivation function SHALL use an approved one-way function such as Keyed Hash Message Authentication Code (HMAC) [FIPS 198-1], any approved hash function in SP 800-107, Secure Hash Algorithm 3 (SHA-3) [FIPS 202], CMAC [SP 800-38B] or Keccak Message Authentication Code (KMAC), Customizable SHAKE (cSHAKE), or ParallelHash [SP 800-185]. The chosen output length of the key derivation function SHOULD be the same as the length of the underlying one-way function output.

The salt SHALL be at least 32 bits in length and be chosen arbitrarily so as to minimize salt value collisions among stored hashes. Both the salt value and the resulting hash SHALL be stored for each subscriber using a memorized secret authenticator.

For PBKDF2, the cost factor is an iteration count: the more times the PBKDF2 function is iterated, the longer it takes to compute the password hash. Therefore, the iteration count SHOULD be as large as verification server performance will allow, typically at least 10,000 iterations.

In addition, verifiers SHOULD perform an additional iteration of a key derivation function using a salt value that is secret and known only to the verifier. This salt value, if used, SHALL be generated by an approved random bit generator [SP 800-90Ar1] and provide at least the minimum security strength specified in the latest revision of SP 800-131A (112 bits as of the date of this publication). The secret salt value SHALL be stored separately from the hashed memorized secrets (e.g., in a specialized device like a hardware security module). With this additional iteration, brute-force attacks on the hashed memorized secrets are impractical as long as the secret salt value remains secret.

5.1.2 Look-Up Secrets

A look-up secret authenticator is a physical or electronic record that stores a set of secrets shared between the claimant and the CSP. The claimant uses the authenticator to look up the appropriate secret(s) needed to respond to a prompt from the verifier. For example, the verifier may ask a claimant to provide a specific subset of the numeric or character strings printed on a card in table format. A common application of look-up secrets is the use of "recovery keys" stored by the subscriber for use in the event another authenticator is lost or malfunctions. A look-up secret is *something you have*.

The CSP SHOULD send a notification of the event to the subscriber. This MAY be the same notice as is required as part of the proofing process.

Replacement of a lost (i.e., forgotten) memorized secret is problematic because it is very common. Additional "backup" memorized secrets do not mitigate this because they are just as likely to also have been forgotten. If a biometric is bound to the account, the biometric and associated physical authenticator SHOULD be used to establish a new memorized secret.

As an alternative to the above re-proofing process when there is no biometric bound to the account, the CSP MAY bind a new memorized secret with authentication using two physical authenticators, along with a confirmation code that has been sent to one of the subscriber's addresses of record. The confirmation code SHALL consist of at least 6 random alphanumeric characters generated by an approved random bit generator [SP 800-90Ar1]. Those sent to a postal address of record SHALL be valid for a maximum of 7 days but MAY be made valid up to 21 days via an exception process to accommodate addresses outside the direct reach of the U.S. Postal Service. Confirmation codes sent by means other than physical mail SHALL be valid for a maximum of 10 minutes.

6.1.3 Binding to a Subscriber-provided Authenticator

A subscriber may already possess authenticators suitable for authentication at a particular AAL. For example, they may have a two-factor authenticator from a social network provider, considered AAL2 and IAL1, and would like to use those credentials at an RP that requires IAL2.

CSPs SHOULD, where practical, accommodate the use of subscriber-provided authenticators in order to relieve the burden to the subscriber of managing a large number of authenticators. Binding of these authenticators SHALL be done as described in Section 6.1.2.1. In situations where the authenticator strength is not self-evident (e.g., between single-factor and multi-factor authenticators of a given type), the CSP SHOULD assume the use of the weaker authenticator unless it is able to establish that the stronger authenticator is in fact being used (e.g., by verification with the issuer or manufacturer of the authenticator).

6.1.4 Renewal

The CSP SHOULD bind an updated authenticator an appropriate amount of time before an existing authenticator's expiration. The process for this SHOULD conform closely to the initial authenticator binding process (e.g., confirming address of record). Following successful use of the new authenticator, the CSP MAY revoke the authenticator that it is replacing.

6.2 Loss, Theft, Damage, and Unauthorized Duplication

Compromised authenticators include those that have been lost, stolen, or subject to unauthorized duplication. Generally, one must assume that a lost authenticator has been stolen or compromised by someone that is not the legitimate subscriber of the authenticator. Damaged or malfunctioning authenticators are also considered compromised to guard against any possibility of extraction of the authenticator secret. One notable exception is a memorized secret that has been forgotten without other indications of having been compromised, such as having been obtained by an attacker.

Suspension, revocation, or destruction of compromised authenticators SHOULD occur as promptly as practical following detection. Agencies SHOULD establish time limits for this process.

To facilitate secure reporting of the loss, theft, or damage to an authenticator, the CSP SHOULD provide the subscriber with a method of authenticating to the CSP using a backup or alternate authenticator. This backup authenticator SHALL be either a memorized secret or a physical authenticator. Either MAY be used, but only one authentication factor is required to make this report. Alternatively, the subscriber MAY establish an authenticated protected channel to the CSP and verify information collected during the proofing process. The CSP MAY choose to verify an address of record (i.e., email, telephone, postal) and suspend authenticator(s) reported to have been compromised. The suspension SHALL be reversible if the subscriber successfully authenticates to the CSP using a valid (i.e., not suspended) authenticator and requests reactivation of an authenticator suspended in this manner. The CSP MAY set a time limit after which a suspended authenticator can no longer be reactivated.

6.3 Expiration

CSPs MAY issue authenticators that expire. If and when an authenticator expires, it SHALL NOT be usable for authentication. When an authentication is attempted using an expired authenticator, the CSP SHOULD give an indication to the subscriber that the authentication failure is due to expiration rather than some other cause.

The CSP SHALL require subscribers to surrender or prove destruction of any physical authenticator containing attribute certificates signed by the CSP as soon as practical after expiration or receipt of a renewed authenticator.

6.4 Revocation and Termination

Revocation of an authenticator — sometimes referred to as termination, especially in the context of PIV authenticators — refers to removal of the binding between an authenticator and a credential the CSP maintains.

CSPs SHALL revoke the binding of authenticators promptly when an online identity ceases to exist (e.g., subscriber's death, discovery of a fraudulent subscriber), when requested by the subscriber, or when the CSP determines that the subscriber no longer meets its eligibility requirements.

The CSP SHALL require subscribers to surrender or certify destruction of any physical authenticator containing certified attributes signed by the CSP as soon as practical after revocation or termination takes place. This is necessary to block the use of the authenticator's certified attributes in offline situations between revocation/termination and expiration of the certification.

Further requirements on the termination of PIV authenticators are found in FIPS 201.