What does "MFA" mean?

Jeffrey Goldberg

1Password
jeff@1Password.com

November 26, 2018

Abstract

Multi-factor authentication (MFA) is widely recommended to people as a measure they should adopt to improve their security. Yet the security that it provides varies dramatically from system to system. Furthermore the additional security resulting from MFA is not always what users expect. For example, users may expect that the security comes from involving different sorts of factors, while the actual security of a scheme may arise from the use of an unguessable and secret which is never transmitted. In other usages apparent MFA may be used for account reset, thus making account compromise easier.

Misunderstandings of the security properties offered by any particular MFA system may lead people to engage in dangerous behavior that they otherwise wouldn't enage in. Perhaps the most dangerous misconception we've encountered is the belief that MFA makes it safe to handle secrets on a compromised computer as long as some other authentication factor remains uncompromised.

In the case of an encryption based password manager, where authentication plays only a minor role in the system's security, offering MFA may lead people to use weaker master passwords than they otherwise would. Thus users may strengthen a less crucial part of their security (authentication), while weakening a far more important component (the master password used to encrypt their data).

We argue that when offering MFA it is important to understand what the often subtle but actual security properties are. Those benefits need to weighed against the security damage that may result

from behaviors that MFA usage might encourage. Additionally, MFA should be presented to users in ways that do not invite dangerous misconceptions.

1 Introduction

Multi-factor (or Two-factor) Authentication (MFA or 2FA) is widely recommended to people as a measure they should adopt to improve their security. And in our experience as the makers of the password manager, 1Password, many people have come to insist on it.

The demand for MFA appears disproportionate to the security it offers in many circumstances. Our experience over many years in attempting to explain our resistance to offering MFA has resulted in numerous conversations in which we attempt to understand why people insist on MFA.

There are real and legitimate security benefits to MFA in many circumstances, but those benefits may rarely be what people are seeking or expecting from MFA, and the particular security benefits may not always be applicable to the individual use case. Among the misconceptions we've observed, the most dangerous is the belief that MFA makes it safe to handle secrets on a compromised computer as long as some other authentication factor remains uncompromised.

We start out with the discussion of the security properties one might want from any authentication system and some of the distinctions between authentication and encryption. These distinctions and security properties are subtle and are probably not well-understood by even by many experts. We then talk about our experiences and discussions with customers, from which we've gained some idea of why people want MFA and what they believe it does for them. Finally, we call for research on expectations of MFA so that we do not have to rely on our intuitions and anecdotes.

2 Authentication desiderata

Authentication, very very roughly, is the process of proving ones identity to some system or service. That service will either accept the proof or reject it. If accepted, it will then grant access to some resource. In what follows I'm going to use terms like "client" and "server" with a great deal of im-

precision. The client may narrowly be something like a computer program such as a web-browser or might, more broadly, be the user in combination with the web-browser.

The most familiar form of authentication for the context of this discussion is providing a password. A user provides an identifier (a username) and a password to their client software (a web browser) which then transmits this information to a server. The server performs some check to determine whether the password is correct for that identity, and grants access iff it is correct.

MFA is (largely) intended to bolster the security of authentication. Before we can assess whether it does and whether it does so in ways that people expect it to, we need to to consider some of the security properties we may seek in an authentication protocol.

CI *Proof of client identity*
: The client should prove to the server that it is who it says it is. In normal password authentication this is done by the client transmitting a secret, the password, to the server.

SI *Proof of server identity*
: For most website logins this is not done during user authentication. Instead, we typically rely on the TLS/X509 infrastructure to authenticate websites to the browser. If the browser accepts the authenticity of the site it may display the little handbag[1] symbol, "🔒". If it rejects the authenticity, it will attempt to alert the user in some meaningful and practical way.

  Mutual authentication is combination of CI and SI.

ZK+ *Zero-knowledge* (*and more*)
: There are a number of technically distinct security properties that together mean that during the authentication process

  1. No secret is sent to either party
  2. An eavesdropper can't learn anything that would help them replay a log in session

[1]The handbag symbol might appear as a closed padlock to many individuals. Felt et al., 2016

3. An attacker who can tamper with the communication can only prevent successful authentication but cannot learn any secrets.

Simple password authentication fails in multiple ways. It reveals the user's secret both to the server and to any eavesdropper.

BigH *Unguessable long term secrets*
The long term secrets (passwords in the typical case) should not be guessable but instead should be high entropy secrets.

This is not generally true of human created passwords. It is, however, true of MFA schemes such as TOTP (see §3.2).

Uniq *Unique secrets*
If an attacker obtains a client authentication secret used for service S, that secret should not be reusable on any distinct service S'. Password reuse is a failure to achieve this.

NoCrk *Uncrackable server verifiers*
If server verification data (e.g., password hashes) are acquired, the attacker should have no feasible way to recover user long term authentication secrets (e.g., the passwords). Passwords stored in plaintext, TOTP secrets, and symmetric PAKE verifiers provide those secrets immediately.

Hashed passwords and asymmetric PAKE verifiers can be used to verify password guessing attempts can all be useful in password cracking attempts.

1Password's Two-Secret Key Derivation (AgileBits, 2017) and Tarsnap's key files (Lucas, 2015) are examples where nothing crackable is stored server-side.

Notif *User notification of login attempts*
User is made aware of (failed) login attempts. Systems like Duo Security or SMS based MFA provide this. (See §3.3)

FS *Secure if some factor compromised*
This is the definitional characteristic of MFA. An MFA system remains secure as long as at least one factor is not compromised. But it must be noted that it is only the *authentication* process that is protected. (cf. CSec)

Reset *Recoverability*
: If the user loses/forgets their long term secrets, some recovery/reset procedure may be desired. Often times this comes at a very strong security cost in other respects, and so must be weighed carefully.

Alt *Alternative authentication*
: Instead of requiring both factors to authenticate, either factor on its own is sufficient in case the user has lost one of them. This is closely related to Reset, and brings with it notable risks, such as account take-over through discovery of answers to "security questions".

Use *Reliably and secrurely usable*

There are some security properties that people may want which are not actually properties of authentication. That is, they are beyond to scope of authentication, but people may none-the-less consider them to be part of authentication:

SSec *Defend server-side non-authentication secrets*
: If Molly (a dog) stores location of her bones on some service, and that service is fully breached (perhaps an insider attack) then that data is compromised even if the authentication process for the service isn't. End-to-end encryption is generally the most effective defense against this threat, but it is not something that authentication schemes can provide.

CSec *Defend client-side non-authentication secrets*
: If the computer that Molly is using to manage the location of her buried bones is fully compromised, then any of those secrets available on that device will be available to the attacker no matter how many factors the authentication system uses. Contrast this with FS.

There are other security properties we may wish of authentication systems, but for this discussion we have listed only a few of them.

3 MFA

It is not necessary to formally define MFA, but it is useful to review how it is typically presented. MFA is often presented as authentication requiring

multiple factors, where the factors are of different types, each type facing distinct threats. Factors types are often described as "something you know," such as a password; "something you have," such as a phone or specific device; and "something you are," such as having a particular fingerprint. Although any of these factors can be stolen or compromised, the security stems from the fact that it requires different attacks and different attacker capabilities to comprise the different types. For example, an attacker who is in a position to capture a password may not be in a position to steal a phone or copy a fingerprint.

There are an enormous number of different MFA schemes, but we will briefly discuss a few which can then serve as reference points for further discussion and also to help illustrate the various security properties.

3.1 Chip & PIN

The ideal example of multiple factors is chip and PIN payment cards. Although it is not what people think about when they consider MFA it illustrate security property FS in a distilled form. Physical possession of the card with the chip is required (something you have) along with the PIN (something you know).

The chip contains a difficult to extract long term hard to guess (BigH) and unique (Uniq) secret. It can prove knowledge of that secret without revealing it (ZK+). The chips make some (fallible) attempts to verify the authenticity of the card reader (SI).

The chip will behave differently depending on whether it has been provided with the correct PIN, thus ensuring that both factors are present at the same time and place. There is also no need for the long term secrets to be stored anywhere other than on the chips, and so this also meets that NoCrk requirement.

If either factor alone is compromised, the authentication process remains secure (FS). An attacker with the PIN sans chip will not be able to fully authenticate; nor will an attacker with the chip sans PIN.

Chip and PIN illustrates MFA at its best. Many of the security properties we might wish of an authentication process are maintained. It is feasible to do this given that it is limited to a specific domain which requires physical presence at a highly tamper-resistant terminal.[2]

[2] An important component of the tamper resistance of the terminals is a result of their

3.2 TOTP

Time-based One Time Passwords (TOTP) will be most familiar to people as what Google Authenticator does, although other client apps are available. When signing up, the server generates a long term unique (Uniq) and unguessable (BigH) shared secret. This is transmitted once to the client typically through displaying a QR code.

On authentication, the client and the server compute a number (typically six digits) that depends on the current time (rounded down to 30 second intervals) and the long term shared secret. If they have the same shared secrets and the clocks are not wildly off[3] they will compute the same code. This will prove to the server that the client has access to that long term secret. No information about the long term secret is transmitted, and replay is limited to a 30 to 90 second period. Thus, this largely has the ZK+ property.

The long term secret is shared between the client and the server, and thus a server compromise means an immediate compromise of that secret. Thus, TOTP fails to meet NoCrk.

Whether or not both factors are truly needed to authenticate is up to the operators of the service. Many services will allow people who have control of the registered email address and access to only one factor to by-pass the factor that they've lost or forgotten. Thus, instead of requiring both factors (FS), these systems will allow authentication through either factor (Alt).

Whether or not Alt is desirable, it is certainly not how the security claims of MFA are typically presented. I, personally, have never heard from a user asking for MFA state that they want it as a way to log in in case they forget their password.

3.3 Push/approval systems

In some schemes, such as those using SMS or Duo Security's product, a message is sent to the user's phone. It may be a one time code to use (as is typical for SMS), or it may require the user to take some approval action. There is a great deal of variety of these systems meeting different security

usage being monitored. There is a high risk to the attacker of being captured when attempting to tamper with the terminal.

[3] I used to manage NTP servers. For me a clock that is off by a few hundred milliseconds is wildly off.

requirements. SMS is particularly terrible for a number of reasons, while other push systems achieve all of the properties that are seen with TOTP. Most notably these also provide notifications to users of authentication attempts (Notif), which can be quite valuable both in terms of real security and in meeting user expectations.

As with TOTP, whether these act as requiring both factors (FS) or as an alternative form of authentication (Alt) is a matter of service policy.

4 1FA

For comparison it is useful to look at the security properties of authentication schemes which tend to be the only factor in single factor schemes or the primary factor in MFA schemes.

4.1 Traditional

The traditional scheme is transmitting a user created password to a server, which, if we are lucky, is storing a hashed version of that password. These only provide client authentication and fail at every other property we may desire.[4]

4.2 PAKE

A Password-based Authenticated Key Exchange scheme (PAKE) provides mutual authentication (CI and SI) without transmitting any secrets during enrollment (ZK+). The protocol involves mathematical computation which cannot be completed unless each party has access to their respective long term secrets.

As it will be necessary for some of the discussion to follow, 1Password's authentication system adds what we call Two-Secret Key Derivation (2SKD) to a PAKE (AgileBits, 2017). This involves a high entropy user secret that is mixed in client-side with their password.

As a very rough summary, Table 1 allows for easy (but approximate) comparison of the security properties of different authentication schemes.

[4]There is one other property, Resistant to Pre-computation Attacks, that simple password authentication has and which is lacking from otherwise far more security

Property	Trad.	TOTP	Push	WebAuthn	PAKE	PAKE+2SKD
CI	✓	✓	✓	✓	✓	✓
SI	×	×	×	✓	✓	✓
ZK+	×	✓	✓	✓	✓	✓
BigH	×	✓	?	✓	×	✓
Uniq	×	✓	?	✓	×	✓
NoCrk	×	×	×	✓	×	✓
Notif	×	×	✓	×	×	×
FS	×	✓	✓	✓	×	?

Table 1: Security properties of selected authentication mechanisms. CI: Client proves identity. SI: Server proves identity. ZK+: No secrets transmitted. BigH: Long term secret is unguessable. Uniq: Long term secret is unique. NoCrk: Server stored verify can't be used for cracking. Notif: User notified of (failed) attempts. FS: Multiple factors.

5 Authentication versus encryption

Before discussing user perceptions, we need to recognize the distinction between passwords used for authentication and passwords used for encryption. Because almost all passwords that people have encountered in their lives have been for authentication users cannot be expected to understand the differences in the security properties between passwords used for authentication and those used for encryption. Sadly, failure to understand that distinction can lead to catastrophic results. (Blaze, 2011).

A password used for authentication is part of the process of proving who you are to some system which accepts the proof or not. That verifier grants or denies access based on its decision. A password used for encryption is the source for a mathematical object that is required to transform data from a useless form into a useful one. There is neither a decision nor granting of access with encryption. Either the password works to transform the data or it doesn't.

For a system which relies entirely on local encryption and does not have any authentication component, MFA is simply non-applicable; after all the "A" of "MFA" stands for "authentication."

schemes. (Jarecki, Krawczyk, & Xu, 2018)

5.1 Password manager unlocking

The distinction between authentication and encryption plays an important role in user's misperceptions of MFA in the context of unlocking data in a password manager.

6 Method

We have made no attempt quantifying demand for MFA discussed in §7. All such statements are based on personal and non-systematic impressions. Also note that we have millions of customers, and so strong demand from a small minority of them will still be a "great deal" from our perspective.

We have made no attempt to count, much less content analyze, the discussion and exchanges mentioned below. We have not even gathered them or pointers to them. The various conversations occurred in our email customer support, on our discussion forums, in comments on our blog, in comments on other people's blogs, on Twitter, and in face to face conversations. We did not attempt to make note of them at the time, and so there is every reason to suspect that what we report here is tainted by selective recall.[5]

All of the individuals we've had these discussions with are people who are already using a password manager or are actively looking at using one. This almost certainly means that they are highly atypical (Stobert, 2014). They are sophisticated network users and are aware of some security issues regarding authentication.

7 Demand for MFA in 1Password

Prior to late 2015, 1Password usage and unlocking did not involve any authentication; it was entirely encryption based. Yet there was a great deal of demand for MFA. In our experience, this demand was motivated by individual's concern for their security, instead of through institutional requirements (*pace* (Cristofaro, Du, Freudiger, & Norcie, 2013)). In a number of

[5]This Selective Quote Method, while not highly respected by experimental psychologists, is accepted by some communities of scholars as long as the paper using the method includes at least one ill-informed reference to Thomas Kuhn and takes pot shots at positivism. Those have been omitted here in the interest of human decency.

reviews of password managers, 1Password was criticized for not offering MFA, despite the fact that there was no authentication component at all to 1Password's security.

At the time, we attempted to explain as best we could why MFA was inappropriate for our security architecture. The success of those explanations was almost certainly limited to the individuals who asked and who were in a position to engage in long technical discussions.

When we introduced our service in early 2016 (public beta in late 2015) two things changed: (1) We actively offered services designed for business-like organization, and (2) we did introduce an authentication component. The demand for MFA became compelling, and some conversations with prospective business customers could be summarized as

"We need you to offer MFA."
"It wouldn't add any meaningful security and would be largely security theater."
"Give us that security theater or we cannot even consider your product."

We have acquiesced (Fillion, 2018).

7.1 Why fight?

Prior to the introduction of an authentication component into 1Password, we were able to say "no" to requests for MFA, as it would have been entirely security theater. Although there may be cases where security theater is justified, we felt it would be harmful here.

The most significant potential for harm is that the use of a theatrical MFA in 1Password might led people to chose weaker master passwords than they otherwise would. This would be a real reduction in data security for no compensating gain in either usability or other aspect of security.

Another way in which MFA can be harmful is that it further confuses the issue of potential reset. We do not have the capacity to reset a master password or account if someone forgets their master password or losses their Secret Key. We attempt to make the clear to users when they first start using 1Password. An MFA factor, however, can be reset, and the existence of something that we can (and do) reset dilutes an important message we are trying to communicate.

Finally, and most obviously, any additional complexity for users or in the code just introduces increased opportunities for things to go wrong.

8 Conversations

Typically when we ask people why they would like us to support MFA their answer is roughly "because it is more secure." If the discussion is occurring in a context in which it is helpful to follow that up, we begin by trying to explain why MFA may add little to the security of 1Password. Prior to launching the 1Password.com service, we would attempt to explain that 1Password works solely by encryption and does not have an authentication component for which MFA could be applied. This is a remarkably subtle distinction, and we do not believe that we have succeeded at explaining it any but the most patient and curious individuals.

With respect to the service, we would attempt to explain that our authentication process is not subject to the same threats that traditional password authentication is subject to. We would attempt to explain the different security properties of traditional password authentication and PAKE+2SKD. Again, we do not believe that we have succeeded in explaining this to all but a few individuals.

During many of these conversations, we explicitly invited individuals to let us know what they were seeking with MFA. We have, indeed, learned from discussions. For example, we had not been *explicitly* aware of Notif as a valuable security property until people pointed it out to use.

Although these conversations took their own individual courses, initial responses roughly fell into three categories:

1. "I want it, and your competitors are offering it", or
2. "It keeps me safe even if my Master Password is discovered", or
3. "It allows me to log in from untrusted computers when I'm traveling."

8.1 More secure

Response 1 typically indicates that the person does not wish to engage in a discussion, and so we shouldn't press them more on their reasons. I personally do not encounter these, as I am usually only brought in to talk with people who do wish to discuss reasons for MFA. I have been told my our Sales team that this is what they most frequently encounter.

People do perceive that MFA systems add security (Cristofaro et al., 2013; Gunson, Marshall, Morton, & Jack, 2011), and so it is no surprise that they wish to have it in a system that is designed to store and manage high value secrets, such as a password manager. I certainly cannot blame people for wanting MFA.

8.2 The scariest error

Response 3 is terrifying. It is a failure to recognize the distinction between FS and CSec. This failure may lead people to view, manage, manipulate, and create high value secrets on a computer that believe may be compromised. It is not surprising that people develop such a dangerously erroneous perception, but it is something that must be corrected at every opportunity.

> Like others have mentioned, if I happen to have a key logger on my computer or if I use a public computer to access my account, my entire account key could be copied by someone. [...] I have 2FA set up on my email account, so I have to authenticate using 2FA any time I'm not at home. [Forum user, January 2017]

And another person in the same discussion offered this as part of their reason for wanting MFA when unlocking 1Password:

> With the speed zero day malware are created these days and with the tools and the many advanced techniques they have available, the frontier is fast shifting where users are at risk almost on a daily basis. This and the numerous corporate intrusions [...] I am not even sure I can trust that my own computer is truly secure despite the fact that it is behind an IDP/Firewall device.

Both of these individuals, as well as others in the same discussion, are clearly knowledgable and thoughtful about their security. Yet they all believe that MFA would protect their use of 1Password on a compromised computer.

In personal conversation an extremely well qualified and experienced security expert explicitly quoted FS to me when trying to explain to me

that MFA does offer CSec. Only after I pointed out that FS only protects the authentication process did they recognize the error.

8.3 Second factor

Response 2 suggests that the person is familiar with the stated rational for for MFA. Yet, like most people, they may not be familiar with threats against authentication mechanisms. If the major threats against traditional password authentication are password reuse and phishing, then someone who is using a password manager may have little to gain from MFA because they have already reduced the threats to their primary factor. This fact, however, will only reduce demand for MFA if it is well understood.

9 Origins

It is not difficult to understand how someone could treat statements of FS as implying CSec. Given how dangerous a belief in CSec may be, it is disturbing to discover that many technically savvy people make this mistake and may even be advising others and spreading this dangerous notion.

The popularity of MFA as a security measure may be in part due its promotion by a number of highly visible services. In some cases, such as Dropbox's introduction of TOTP (Agarwal, 2012), it may be a defense against password reuse by their customers. Recall that TOTP involves a unique and unguessable long term secret.

In other cases, services may be promoting what appears to be MFA as providing alternative forms of authentication. Password reset and account recovery is a notoriously difficult problem, and services may be using the ability to authenticate via one factor (perhaps also with a demonstration of control of the initially registered email address) as sufficient proof of identity to allow for resetting the other factor. Services may be inviting people to believe that these systems increase account security even as they increase the risk of account take-over.

10 A call for research

To my knowledge there has not been research into what sorts of security people expect from MFA. It may be that this is an impossible question as it assumes that people have an implicit or explicit threat model in the first place. However, it may be possible to study the beliefs of the more sophisticated users.

Authentication, the threats to it, and the defenses against those threats are far more complicated and subtle than we can expect most people to understand. Therefore we must build systems which encourage behaviors that are in the users privacy and security interests without the expectation that the user must understand the threats. This doesn't mean that we can't attempt to correct for dangerous misconceptions, such as the belief that using MFA makes it safe to manage secrets on a compromised computer.

I have been jumping to conclusions about users' perceptions of MFA based on conversations I've had over the years. I would like to know whether those conclusions are correct, and I am particularly concerned about those hypothesized perceptions which may be leading people to dangerous behavior.

References

Agarwal, A. (2012, July 31). Security update and new features. Retrieved May 24, 2018, from https://blogs.dropbox.com/dropbox/2012/07/security-update-new-features/

AgileBits. (2017, April 12). 1Password security design. Retrieved from https://1password.com/teams/white-paper/

Blaze, M. (2011). Wikileaking a cryptography lesson. Retrieved July 22, 2014, from http://www.crypto.com/blog/wikileaking/

Cristofaro, E. D., Du, H., Freudiger, J., & Norcie, G. (2013). Two-factor or not two-factor? A comparative usability study of two-factor authentication. *CoRR*, *abs/1309.5344*. arXiv: 1309.5344. Retrieved from http://arxiv.org/abs/1309.5344

Felt, A. P., Reeder, R. W., Ainslie, A., Harris, H., Walker, M., Thompson, C., ... Consolvo, S. (2016). Rethinking connection security indicators. In *Proceedings of the twelfth symposium on usable privacy and security*. Symposium On Usable Privacy and Security.

Fillion, R. (2018, April 25). Multi-factor authentication in 1Password. Retrieved May 24, 2018, from https://blog.agilebits.com/2018/04/25/multi-factor-authentication-in-1password/

Gunson, N., Marshall, D., Morton, H., & Jack, M. (2011). User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. *Computers & Security*, *30*(4), 208–220.

Jarecki, S., Krawczyk, H., & Xu, J. (2018). OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks. Cryptology ePrint Archive, Report 2018/163. Retrieved from https://eprint.iacr.org/2018/163

Lucas, M. W. (2015). *Tarsnap mastery: Online backups for the truly paranoid*. IT Mastery. Tilted Windmill Press.

Stobert, E. (2014). The agony of passwords: Can we learn from user coping strategies? In *Chi'14 extended abstracts on human factors in computing systems* (pp. 975–980). ACM.