Who's at Your Keyboard? Two-Factor Authentication and Trends for Accessing Accounts and Online Resources

By Eric E. Cohen, CPA

Eric is a prolific author, engaged in virtually every effort to standardize accounting and audit data, a national expert on a wide variety of standards efforts, and co-founder of XBRL.

He is a contributing editor to *ThinkTWENTY20*.

As more aspects of our personal and business lives, finances and other confidential information move online, with the catalyst of the Pandemic making in-person contact inadvisable and traditional postal mail becoming more unreliable, we need comfort that only we (or those we authorize) have access to our online information and only we can authorize online activities on our behalf. Beyond login names and passwords, the trend has lately been to the use of an additional authentication method that provides evidence that you are, indeed, you. This article speaks to trends in two-factor authentication (2FA) and multi-factor authentication (MFA), related tools, and challenges associated with the growth of 2FA.

If your email inbox is like mine, you have seen an increasing number of incoming messages with a common theme. They begin "Your sign-in method is about to change" (or, more urgently, "Your sign-in has changed"). That message goes on to say two-factor authentication (or 2-step verification – 2SV – more on that to come) is going to be required, or the requirements will be made more stringent "for your protection."

Facebook, for example, is pushing more and more "at risk" accounts to mandatory 2FA. Google has started requiring 2FA for its users after a few years of recommending it. Twitter has a number of interesting options involving 2FA. Other services that required one form of 2FA raise the bar from older 2FA methods to newer ones.

In this article, we will speak to 2FA: Why is it helpful or necessary? How can it be implemented? What are some of the key advantages and disadvantages of the choices, where some of the best-known options include text messages/SMS, authentication apps and hardware? What's next on the horizon?

Why Is 2FA Helpful, Or Even Necessary?

In the Fall 2021 edition of *ThinkTwenty20*, I wrote about the evolution of payment methods, leading up to the cryptocurrency era. We focused, in particular, on cryptographic key management, a major change from traditional sign-in approaches.

As part of that discussion, I spoke to passwords and how, 30 years after the Web got its beginning in 1991, enterprise breaches disclosing login names and passwords increasingly make the headlines, and passwords alone seem to do more to keep honest people out than criminals. As time moved on, users were told to make their passwords more difficult to guess, forced to change their passwords on a periodic basis and required to include variations, such as numbers, upper and lower case characters and punctuation. The question was whether the cost of inconvenience was compensated for with the benefit of safety; I had a sign-in to my first email account for 20 years before being forced to change the password.

Looking at the emergence of 2FA is the flip side of looking at the failure of password protection and the shortcomings of other methods of proving that you are who you say you are.[1] If we can use other means to reinforce or replace passwords, perhaps we have a chance in the fight for authentication of our own online identity. Three factors the community considers in authentication:

1. Who you are (e.g., using a fingerprint or facial scanner).
2. What you know (e.g., passwords).
3. What you have (your mobile phone, a hardware token).

Two-*factor* authentication will involve two of these three factors. Two-*step* verification will involve at least one of these factors, but in two stages. For example, entering a password and then a pin, or your mother's maiden name, or you favorite musical instrument, are both based on knowledge, so these are examples of two-step verification. Where your need to use your password (what you know) and a fingerprint scan (who you are), however, involves two different factors.

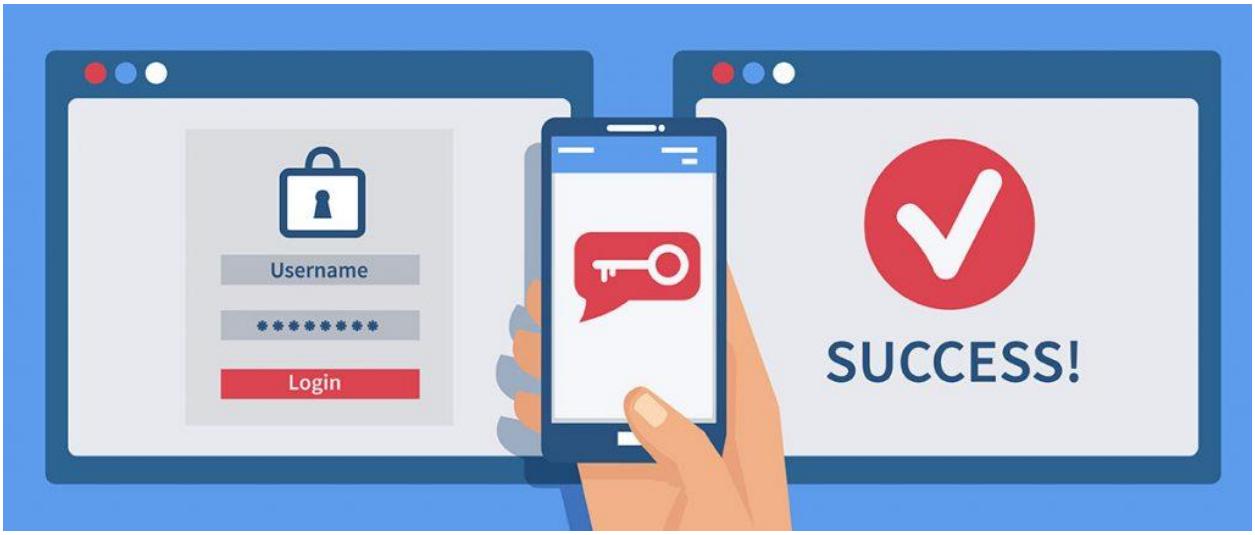
One group dedicated to dealing with the problems of people and their passwords is called the FIDO Alliance, an open industry association started in 2013 with the mission to come up with standards to "help reduce the world's overreliance on passwords."[2] The Alliance shared this information,[3] explaining their passion for replacing passwords:

- Passwords are the root cause of over 80% of data breaches.
- Users have more than 90 online accounts.
- Up to 51% of passwords are reused.

This is probably not news to any of us. As our use of online services began to grow, having and tracking site-specific logins became more difficult. One site requires punctuation marks, another prohibits it. One requires eight characters or more, another limits it to six. We may visit one site daily, another monthly, another annually at most, and the requirements change.

As our use of multiple devices (from PC to smartphone to tablet) began to increase to be able access those services wherever and whenever we wished – as, after all, the Cloud led to the expectation that we can access all of our information and services anywhere, from any device – it made us want easy access from our PC, our phone, our tablet, from the hotel business center … from anything that could access the Internet. Sadly, that also meant an attacker could get at our information and services anywhere and any device if they know what we know and can imitate what we have.

Entering a password was a common ability across all devices, the least common denominator. Although I have a fingerprint reader on all of my devices, they are not synchronized, as my identity is not standardized across my devices. My password managers on my devices are also not synchronized and represent multiple operating systems. Even on my iPad, the Apps don't all go to the same resource for passwords as the web browsers, so each installation or update may require manual steps to try to keep an App and a website from the same organization in synch. This is particularly painful when an App is updated (which may happen infrequently) and the update does not remember the password previously used in the App – which wasn't automatically stored in the device's password manager.



The problem with passwords, in particular the ease of guessing them, isn't lost on government standards setters. In the United State, the National Institute of Standards and Technology (NIST)[4] has been raising an alarm for a long time, such as encouraging users to avoid passwords exposed in data breaches. One popular website used to find such exposed sign-in credentials and passwords is haveibeenpwned.com.[5] The etymology of "pwned" is up to some debate, but whether from mistyping ("You have been owned" becomes "You have been pwned"), taken from chess (where you are treated like a pawn and I have ownership of your login credentials by taking advantage of your weakness or other failure) or some other origin, such as when evil doers can take advantage of failures in security and access your resources, "you have been

pwned." Resources such as Firefox Monitor (from the Firefox browser people) can track your email addresses and warn you as the addresses are found in the latest breaches, leveraging the ihavebeenpwned database.

So, we know that passwords are the easiest and most common safety tool for our sign up, but no amount of requiring users to make them more difficult to guess has stopped the bad guys from getting them. Password security alone has let us down, in some part because we are too lazy to create passwords that are not easily guessed, because enterprises that demand passwords are involved in leaks where our passwords have been exposed as clear text, because we are easily fooled by social engineering and deliver our login credentials to people who pretend to be technical support, and for numerous other reasons. In one situation I find particularly troublesome, people who wish to buy cryptocurrency from the well-known exchange Coinbase, using their bank funds, are asked to provide their login credentials (name and password) for their online bank account to Coinbase for a service called Plaid Technologies.[6] That raises many issues.[7]

Where passwords fail, then, we look to 2FA as an adjunct or alternative. The need to use a fingerprint scan or have your mobile phone at hand may not be perfect, but it raises the bar to get access to your resources.

So, 2FA moves beyond "what you know" (such as a PIN or password) to what you have:

- A phone for voice calls or text messages.
- A device that can receive email.
- Special "authentication software," such as offerings from Google, Microsoft.
- Some other physical object or token that works in conjunction with security systems.

Implementing 2FA

In my last *ThinkTwenty20* article, I noted:

> *Although two-factor authentication (TFA) devices – combining "who you are" (such as fingerprints or eye scanning) or "what you have" (such as a separate piece of equipment) to "what you know" (your login and password) – were widely available 20 years ago (in 2003 RSA SecurID,[8] as a "what you have" device, held 70% of the market as a hardware device to make logging into systems more secure), it has only been the recent highly publicized privacy breeches that have led more information environments to strongly encourage or require TFA. For example, I can access my email or systems for a university with which I am associated only with the use of one of my smart devices as an authenticator, using Microsoft Authenticator[9] for TFA. Cryptocurrency exchanges, such as Coinbase, highly recommend TFA; Coinbase supports Yubico's YubiKey.[10]*

Shared secret via phone, text or email

Where the ability to type a password, was the first and simplest step across devices and locations, the next common denominator for most people who need mobile access to their resources is a mobile phone (not necessarily a smart phone). For that reason, two of the most common ways that services have authenticated users is with a phone call or text. The use of a "shared secret" – a code delivered on demand – is conveyed either with a text message (SMS) or delivered using text-to-speech via a phone call. With the advent of smart phones, email became another common channel by which the shared secret can be sent.

The costs and timeliness of receiving the code varied between these options. The costs are amplified for people who are travelling away from their home. Text messaging is often the most reliable and least expensive of these depending on the user's phone plan.

Some services have yet to move to this level; others consider it as inadequate for some or all of their users. While the most casual thief may be delayed by these methods, they are anything but foolproof. Any access to the phone may offer a view of the code from an email or text message in the notifications area, even if the phone is locked. Hackers have been able to clone the SIM cards, use social engineering to get the phone number changed to one they control,[11] or remotely access voicemail or text systems. One-time passwords can be intercepted or social engineering used to have the user provide the code to the bad guys.[12] This has led to calls that the security minded move from SMS to other methods.[13]

Server-based authentication with a token

For environments where additional security is required (financial services, accounting and others who might be targeted with potential greater risks), the use of a separate security device, often in a design to fit in with keys (keyfob token) or credit cards (credit card token), began to rise.

Twenty-five years ago, Security Dynamics[14] introduced a system called SecureID,[15] part of a system known as ACE/Server.[16] Key to the process was the SecureID device carried by users. It came in different form factors, including a small key fob, a credit sized "standard card" or a PINPad. Each looked a bit like a clock, with a screen displaying a code that updated frequently on it (so even if someone saw the code, it would likely have changed by the time they went to enter it). EY and other CPA firms used these

for additional security in the days before we all carried smart phones. While these are still in use, virtual versions, known as a soft token app, grew in popularity for their convenience.

The special code was generally generated based on an agreement of the time between the device and the service (producing a time-based one-time password, or "TOTP") or based on the hash of the previous password (sounding a lot like blockchain's foundation, a hash-based message authentication code (HMAC)-based one-time password, or "HOTP"). In either case, both are based on secret algorithms and, in a 2011 cyberattack, there was a leak of some of those secrets. At the time, there were 40 million of the hardware tokens and 250M users of the mobile software.[17] In 2021, 10 years having passed and non-disclosure agreements lapsed, many RSA executives bound by non-disclosure agreements have come out to tell the story of the experience.[18]

Dedicated authentication software

Microsoft, Google and others now offer dedicated authentication software that can run on your various devices. Along with the ability to display a time-based, one-time password for the user, they themselves can communicate the OTP with servers autonomously to authorize a connection. For example, when I need to access the resources at Bryant University using my laptop, such as my email or my class schedule, it sends a signal to my Microsoft Authenticator on my mobile device, which I can authorize or ignore. On my mobile device, it works seamlessly, so I don't have to go through the second action of having to click a few buttons to authorize the use. Google Authenticator is another option, and can work with Office 365 with the right settings.[19] Likewise, Microsoft Authenticator can be used with Google's product suite.[20]

YubiKey and physical security keys

Physical security keys offer a level of security and flexibility beyond that of a software authenticator. Having mentioned the work of the FIDO Alliance previously, the development of Universal 2nd Factor (UTF) means you can access any number of online services using a single security key instantly and with no need for loading drivers or special software.[21] As long as the service is compatible with FIDO and UTF,[22] access is a snap, in theory. Unlike a password or authentication app, delegating access to someone else is as easy as handing them a copy of your key. Making multiple keys is simple at setup, and very difficult afterwards.

While the Google Titan security key and other third parties are in the market, YubiKey from Yubico[23] is an acknowledged leader in the field. Services that support the YubiKey include the major platforms from Microsoft, Google and Amazon; social media sites Reddit and Twitter; and cryptocurrency exchanges Coinbase and Binance.[24]

To provide a review of the YubiKey, I received two of the different models in the YubiKey 5 series.

Obviously, there is no keyhole in your devices – only USB-A (older “desktop” USB), USB-C (the newer oval shaped port) and Lightning (common to iPhones and iPads). While Bluetooth is not available for YubiKey, NFC (near-field communication, available on most Android phones) is available.

Testing on a laptop with USB-C, an iPad with Lightning and two Android phones (each with NFC, one with micro-USB and one with USB-C), I tested two units, each of which looks a bit like a flash drive:

- YubiKey 5Ci: works with USB-C and Lightning (so, laptop and iPad).
- YubiKey 5C NFC: works with USB-C and NFC (so, laptop and Android through NFC).

There are also keys with higher levels of security, including additional biometric checks.

For my own safety, I will not name the accounts for which I tried to set up the YubiKey. The online instructions on how to set up the key for each service was complete, but the process was not always intuitive. You go to the service you wish to associate with the key; find the security settings page; find the option to associate a security key; and follow the instructions to insert the key and tap the device. Part of the challenge is that the services themselves may have only partially adopted the standards on which the YubiKey is based. In particular, mobile versions/Apps may be behind desktop adoption. If I access the service on my iPad through the App, it uses the traditional security, but on the same device, through a browser, it requires the YubiKey.

Once the setup is done, accessing the service with the key is relatively painless. When you go to log in, you are prompted to insert the key or place it near the NFC receiver. A light on the device will start to blink, indicating you need to interact with the device – by putting your finger on a gold disk in the middle of the key (5C NFC), or metal bumps on the side of the key (5Ci). After being plugged in and tapped, it does the rest for you.

For those of us who enjoy the convenience of any device, anywhere, having to keep a key at hand appropriate to your device is a bit of an inconvenience, but so is dealing with the “robust” passwords suggested by password managers as you move between devices. There is more of a learning curve in the setup than I anticipated. When you read about account takeovers, however, and the problems people have with other methods, these durable units that require no batteries have a bright future as long as services support them. Few financial institutions do at this point in time.

What's Next on the Horizon?

Where do we go from here? How do we get past the compromises between confidence and convenience? How do we get past the groups that say global identity systems will make all of this go away, while others are concerned that global identity systems will be misused and cause far more problems than they are worth?

In the future, private and public key technology could be used as part of facilitating the secure exchange of configuration and credential data between multiple user devices and provide end-to-end security.

My own exploration into this area began with the early days of XBRL, as we wanted to consider how to bring the auditor's report into the electronic age and have a trustworthy auditor's signature on the report. In my collaboration with groups like the World Wide Web Consortium on digital signatures and XML encryption, I had the opportunity to meet a gentleman named Phillip Hallam-Baker.[25] Our names jointly appear on W3C XML security standards.[26]

Catching up with Phillip a few years ago, he told me that he was working on an idea called "The Mathematical Mesh."[27] He told me that, in his dealings with Sir Tim Berners-Lee[28] (known as the "inventor" of the World Wide Web), Sir Tim had tasked him with coming up with a better answer to accessing web resources than logins and passwords, but only 25 years later was he able to get around to it. In his vision, private and public key technology (as used in blockchain and cryptocurrencies) would be used as part of facilitating the secure exchange of configuration and credential data between multiple user devices and provide end-to-end security.[29]

Where the Mathematical Mesh will go is a good question. What is obvious is that, if cryptocurrencies continue to grow in popularity, the need to efficiently deal with wallets and keys, as well as with online accounts, will amplify the demand for safe and user-friendly tools.

Final Thoughts

Twenty-five years ago, our information and accounts began migrating to the World Wide Web in earnest. Password requirements expanded from allowing MyCatFluffy to looking like a stream of epithets from Mad Magazine ("#$5-66yY-@Tty") and access to accounts required having the account holder's text or email to login. Google and others tell us that even that's not enough. Why can't we just put the car in the ignition and move forward? And does the demand for the convenience to start the car remotely mean the key needs to move back to the phone?

Whichever direction this goes. tools like Yubico's YubiKey may be an important part of the solution.

[1] https://workos.com/blog/a-developers-history-of-authentication.
[2] https://fidoalliance.org/overview/ - FIDO stands for “Fast IDentity Online”.
[3] https://fidoalliance.org/what-is-fido/.
[4] https://www.nist.gov/identity-access-management.
[5] https://haveibeenpwned.com/.
[6] https://plaid.com/how-it-works-for-consumers/.
[7] Along with giving someone else your password, which you should never do, a different abuse of the relationship was evidenced in a recent class action lawsuit, resulting in a $58 million settlement about data privacy issues. https://fingfx.thomsonreuters.com/gfx/legaldocs/dwpkrgbdrvm/Plaid%20settlement%20memo.pdf.
[8] https://www.rsa.com/content/dam/en/data-sheet/rsa-securid-hardware-tokens.pdf.
[9] https://docs.microsoft.com/en-us/azure/active-directory/user-help/user-help-auth-app-overview See also Google Authenticator at https://www.google.com/landing/2step/.
[10] https://help.coinbase.com/en/coinbase/getting-started/getting-started-with-coinbase/2-factor-authentication-2fa-faq.
[11] https://www.techrepublic.com/article/two-factor-authentication-cheat-sheet/.
[12] https://krebsonsecurity.com/2021/09/the-rise-of-one-time-password-interception-bots/.
[13] https://www.pcmag.com/opinions/leave-sms-authentication-behind-get-an-authenticator-app.
[14] https://www.youtube.com/watch?v=Luqi12VWB3o.
[15] https://en.wikipedia.org/wiki/RSA_SecurID.
[16] http://media.corporate-ir.net/media_files/NSD/RSAS/ar_1997.pdf.
[17] https://www.cnet.com/tech/services-and-software/rsa-cyberattack-could-put-customers-at-risk/.
[18] https://www.wired.com/story/the-full-story-of-the-stunning-rsa-hack-can-finally-be-told/.
[19] https://docs.microsoft.com/en-us/microsoft-365/admin/security-and-compliance/set-up-multi-factor-authentication?view=o365-worldwide.
[20] https://support.microsoft.com/en-us/account-billing/add-non-microsoft-accounts-to-the-microsoft-authenticator-app-7a92b5d4-d6e5-4474-9ac6-be0b6773f574.
[21] https://www.yubico.com/authentication-standards/fido-u2f/.
[22] https://www.ftsafe.com/article/620.html.
[23] https://www.yubico.com/.
[24] https://www.yubico.com/works-with-yubikey/catalog/?series=3&sort=popular.
[25] https://en.wikipedia.org/wiki/Phillip_Hallam-Baker.
[26] https://www.w3.org/TR/xmlenc-core1/.
[27] https://mathmesh.com/.
[28] https://en.wikipedia.org/wiki/Tim_Berners-Lee.
[29] https://www.ietf.org/id/draft-hallambaker-mesh-cryptography-08.html.