Zeropasswords

Patented

A digital identity that cannot be stolen

White paper

Nikos Leoutsarakos

Tiny bio

Nikos has a Physics background and a M.Sc. in Computer science from McGill University in Montreal, Canada, where he lives with his wife and two children. He has been publishing and developing systems in the areas of cryptography, wireless, mobility and digital signatures since 1994..

Nikos can be contacted at NIKOS@ZEROPASSWORDS.COM

Prelude

Identifying the cause is halfway to a cure. The cause of all our Internet breaching troubles is the membership paradigm. In our natural life, if one wants to belong to a group he must become a member. In the digital world, if one wants to access a Website, an online service or game, a cloud, or a server, he must also become a member, i.e. become a registered user. And what do Websites do? They keep lists of their members and proofs of their membership in databases, the same databases hackers take.

Today, after more than 50 years, millions of breaches (hackers on the news often) and millions of databases stolen, we have ample evidence that while the membership paradigm may work in real life, it sure does not work online. It does not work online because users are obliged to share their secrets with Websites, servers and clouds, and every time a secret is shared, one of the parties eventually leaks it either intentionally or unintentionally. It is time to start looking for other ways to belong to online groups besides "membership by a shared secret". Perhaps there is a way to belong to a group without being a member in the traditional sense, i.e. be recognized by a secret that you know or by a secret that you have. We tried that and it didn't work because the Internet is not good at keeping secrets.

Here is an idea. Why be a member if you can be a relative? Yes, I am talking about your relatives, my relatives, anybody's relatives, those folks that are in your family whether you want them or not. Relatives belong to a group (an extended family) without sharing a secret. They do not have an attribute, privilege, title or a secret that they obtain or gain from a source or from an organization in order to become a member of a family. Relatives are bound by blood, an attribute which they can never lose, or be taken away from them, because it is not external to them, it *is* them. Isn't that grand? Relatives are bound not by something they possess, or by something they have memorized, but by who they are!

There is a lesson somewhere in there.

Lesson

Your relatives will be your relatives for ever! If you like them you are lucky and there is no problem. But if you don't, I am afraid the only way to get rid of them will cost you life-in-prison. What binds you to your relatives is "blood", a bond that neither you nor they can ever lose because it is in you, it *is* you. Not only you can never lose your relation to your family but others can never acquire it. Well, wait a minute. What if I marry into a family? Don't I automatically become a member of the family?

The distinction between *blood-relatives* and *relatives-by-marriage* is the lesson here.

While blood relatives are bound by "who they are", relatives by marriage are bound by a certificate. A marriage certificate that is. For them, relation to the family is an acquired attribute that society provided to them in the form of a document, and as easily at any moment, society can take this attribute away from them with another document: a divorce certificate! Conversely, blood relatives don't need a certificate to belong to a family. They just belong! Blood relatives are not afraid of the other relatives, the society or anyone else for that matter, to ever expel them from the family because no one can change them; no one can change who they are. It's just not possible.

Today, we are all "relatives by marriage" to cyber groups, such as Websites, clouds and online services. Take Websites for example. They generate digital identities for us "visitors" in the form of digital profiles, tell us which part of the profile is the secret that we need to provide every time we login, and then warn us of the consequences of forgetting or losing that secret. Doesn't that make you feel like a relative-by-marriage to Websites? It does to me, and I don't like it anymore. I want to be a "blood relative". I want to control my membership. I want the control out of the hands of the Website and into my hands and the hands of registered users. The point I try to make is that in today's login systems, regardless of how you authenticate yourself, either by something you know (password), or by something you have (security token), or even by something you are (fingerprints, retina, your face), it is the cyber group (Website) that will verify and judge your membership and ultimately decide whether you are allowed to login or not. If cyber groups get to decide, as they do today, you are not a "blood-relative".

Imagine for a second that we could replace today's cyber groups of "relatives by marriage" with cyber groups of "blood relatives". How would the Web be then? Imagine a simpler, friendlier Web where registered users no longer use a certificate, such as a password or a token, to prove their membership to a Website. Instead, registered users are members of Websites because Websites *need* them! Try to envision a cyber-world where some Websites need you and some don't, which in today's terminology translates to, being registered at Websites that need you, and not being registered at Websites that don’t need you[1]. The question is: why would a Website need its registered users? For what? What could users possibly have that a Website needs?

Suppose you need to access your files at a Website but the Website has no idea where your files are and needs you to reveal or specify the directories and URIs of your files. Doesn’t that Website need you in order to retrieve your files? Or suppose you want to login to a Website but the Website does not have your profile to log you in because you took it away from the Website! Doesn’t that Website need you to bring your profile back so that it can log you in? Or suppose you want to play an online game but the game does not run because it is missing a configuration file or a module that you took away from the game server. Doesn’t that game server need you to bring back the module or the configuration file to complete the game so that it can run? Do you see how registered users could control their memberships at various Websites and online servers (or services) by removing vital parts that Websites and servers need to function and provide them later at will? Wouldn’t you agree that if registered users could do that successfully and efficiently they would have finally become “blood relatives” and for the first time stop being in the mercy of Websites?

We start our journey to discover more about this new way to login, --this new way to become a member and control your membership and the positive impact it will have on our digital lives, by pointing out the crucial role *representation* plays in software apps.

Representation

Software is not real. It is a representation of real things. Developers put it together out of pixels and sound waves (for now). The best way to think of an app is to think of it as a slice of human experience. Software apps are representations of our experiences from interacting with the physical world, intellectual world, and from interacting with one another. For example, we capture typing in a word processor, picture taking in a frame grabber app, listening to music in an MP3 app, or payment in a credit card processing app. Success of an app is intimately linked to the clarity of its representation of the real thing. But be careful. The representation of an app is not its GUI, nor is it the set of its operations, its features or capabilities. Representation is an invention, a discovery that marries form and function seamlessly.

A good example of the importance of representation in software is sending and receiving documents electronically. A few years ago we had two options: fax a document using a fax

[1] A Website does not need you because it has no members. It is a public Website, open to everyone (no login)

machine or use a fax application. Every time you needed to send a fax from your computer, you had to activate a fax application, use it to send the fax, and close it. One day, someone decided to develop a printer driver that performed the faxing function. He called it, fax driver. With the fax driver installed, users were now able to send a fax from within any app. For example, a secretary who has just finished typing a letter could use the word processing app to send it to a printer and if needed fax it to a phone number, all without exiting the app. All she had to do is change drivers. The representation of the faxing activity as a fax driver effectively killed the standalone fax app and wiped out all faxing software products from the marketplace! Companies went bankrupt and lost their investments, all because of a shift in representation. Fax machines survived this shift in representation and continued to sell. But not for long. A few years later we changed our representation of sending and receiving documents again. Instead of faxing documents to one another using fax machines, we decided to scan documents and attach them to email or instant messages and use printers at the receiving end to get them on paper. We no longer needed fax machines, fax drivers and fax protocol standards. What we needed instead were printers, scanners, scan drivers and message transfer protocol standards. This had a major impact to fax machine manufacturers. They had no choice but to stop producing unwanted fax machines and start producing printers and scanners with built in connectivity. Interestingly, our need to send and receive documents had not changed since the beginning. It is our representation of this activity that changed and it will continue to change in the future always enhancing our experience.

The prevailing software representation of our login experience today is *verification*.

Verification of secrets to be exact. We chose to capture and represent our login experience as a verification system that compares a secret provided by one party against secrets stored by the verifying party. If verification succeeds, i.e. the provided secret matched one of the stored secrets, login succeeds and access is granted. If no match is found, login fails and access is denied.

As you can see in Figure 1, in order for a secret-verification system to function properly, each party must be responsible for the safekeeping of its own secrets. Alice is responsible for the maintenance and safekeeping of all the secrets others have given her and of all the secrets she has given to others. Bob is responsible for his secrets too. In reality, keeping the secrets secret has been proven very hard to do.

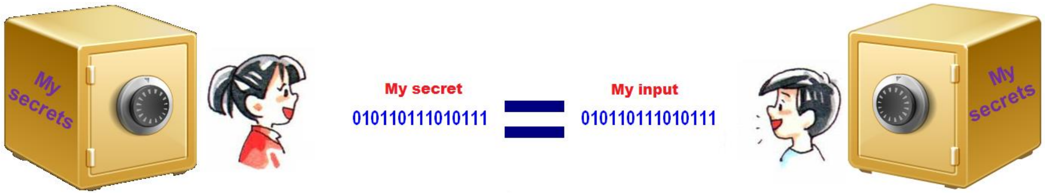

Figure 1

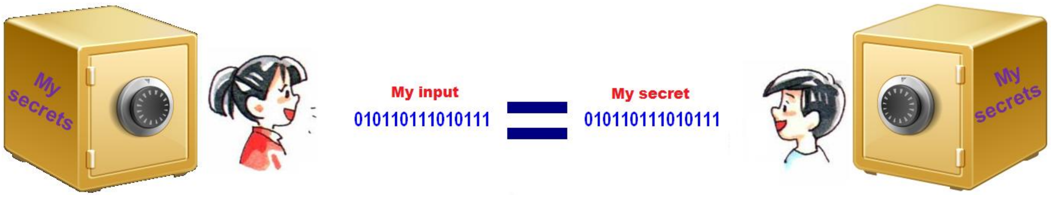

Figure 2

Also, examining Figure 1 closer you may notice that a secret-verification system is one-sided because only one of the two participating parties can perform the verification process. Notice that in Figure 1, Bob tries to gain the trust of Alice (verifier) by proving his knowledge of her secret and in Figure 2 Alice tries to gain the trust of Bob (verifier) by proving her knowledge of his secret. If two parties want to trust each other they need to perform two separate verifications[2].

Belief

Representing login as verification is not an accident. It is the product of our belief that two parties can trust each other if they both know a common secret. This belief is about 2,500 years old, when the ancient Greek and Roman military first used shared secrets, passwords, to identify comrades. About 50 years ago we adopted this antiquated technology to provide timesharing for large mainframe computers and we continue to use it today to provide security online.

Precisely because we believe that trust online stems from the knowledge of shared secrets, we built and we continue to build login systems that verify secrets. Implementations of access control (login) systems today vary in the process they use to verify secrets and vary on the type of secret they are capable of verifying, but they do not differ in their representation of the login activity, and without exception, all login systems today are secret-verification systems.

Password-based login systems verify passwords against secret password samples; fingerprint-based login systems verify fingerprints against secret fingerprint samples; token-based login systems verify generated tokens against secret token samples; PIN-based login systems verify PINs against secret PIN samples stored in smart cards; even retina-based login systems verify

[2] User always tries to gain the trust of a Website in order to gain access. It doesn't get more one-sided than that!

retina patterns against secret retina pattern samples. Without realizing it over the years, Websites and online servers have accumulated billions of secrets in databases all over the Web, and now we try desperately to keep all those stored secrets and samples safe in an ongoing battle with hackers.

If we want to see a change in login systems we need to change our belief about trust on the Web and about online trust in general. We must stop believing that *sharing-a-secret* is the only way two parties have to trust each other, and start seeking another way, another reason, for two parties to trust each other. Once we have a new belief, it will give birth to a new software representation of our login experience, which in turn will spawn a new breed of login systems.

Mutual Need

I started a quest for a "new belief" for trust on the Web by putting myself in the shoes of one of the parties (Figures 1&2). I asked myself: besides repeating back to me one of my secrets, what else could the other party give me, say, or do to make me feel comfortable enough to trust it? My answer to this question after some research and personal soul searching is *nothing*. There is nothing a cyber-party can give me, or say to me, to make me trust it and truly believe everything it says, claims, or does. For all I know, the other party may be a person, even a nice person, or it may be a system, perhaps a legitimate system, but unfortunately it appears to me as a bunch of bits transmitted from the other end of the Internet and I am not wired to trust something, or someone, that I cannot see, hear, feel or touch. I am also certain the other party feels the same about me.

After feeling hopeless for a while, I was reading an article one day that referred to the Alan Turing test. This conceptual test that computer scientists learn at school determines a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. "If you have no physical contact with the other party", Alan Turing said[3], "and the machine behaves intelligently you have to accept the fact that machines can think". And because I was biased thinking about online trust for quite some time, I heard these words of Alan Turing differently this time and a question popped up. Why couldn't we extend the Alan Turing test to not only determine whether someone is a human or not, but to also determine someone's online identity without physical contact? What if someone's online identity was a unique behavior or a unique ability that could be demonstrated online? Then, we could determine someone's identity by someone's unique ability to perform a task, or make an event happen. Here are some exaggerated examples. If someone claims to be Moses, we could ask him to part the waters; if someone claims to be Jesus, we could

[3] Paraphrased

ask him to turn water into wine; if someone claims to be King Arthur we could ask him to remove the sword from the stone; and if someone claims to be Frank Sinatra we could ask him to sing.

“Identity by unique ability” sounded like a good candidate to replace “identity by shared secret” and I decided to pursue it further. A party’s identity would be determined by a party’s ability to perform something unique, or perform something uniquely? That would work but it would be one-sided. Could we avoid running the test twice in order to determine the identity of both parties? Could we devise a reciprocal online identity test? It turns out that all we have to change is to make the accomplished action, task or event require both parties? What if King Arthur was unable to remove the sword from the stone without the help of his future wife and queen? A successful removal of the sword from the stone in that case, would automatically prove two identities at once: the identity of the king and the identity of the queen.

I propose to replace our belief that trust stems from *verification of shared secrets* between two cyber parties, with the new belief that trust stems from *mutual need* between two cyber parties.

If we can have two parties needing each other rather than sharing a secret, we will have given them a new way, a new reason, to trust each other. Traditionally, when we share a secret with someone, we ask him to repeat the secret back to us, and if he does we feel good, and this good feeling helps us trust him. As previously noted, this is one-sided and it doesn’t help him; it only helps us to trust him. In the proposed new way, we ask someone to cooperate with us to make an event happen. If he does, and the event indeed happens, we feel good, he feels good, and our reciprocal good feelings help us both trust each other. If in addition, we and the other party both benefit from the outcome of the event, the feeling becomes especially gratifying and both parties trust each other more.

Here is the mind shift in a sentence: two parties trust each other because they need each other. In other words, both parties need the outcome of an event and neither party can have it unless both parties cooperate to make the event happen. Hence their (mutual) need of each other. One can see how this could be extended to prove the identity of, and trust several parties at once by requiring everyone’s participation to make an event happen.

Looking at Figure 3 an obvious question comes to mind: what if ten different parties besides Bob can help Alice make an event happen via the same enabler? How can Alice then tell who the other party is? Clearly she can’t. What is required is an enabler that can guarantee to Alice that out of billions of cyber parties, there exists one and only one party, Bob, who can help her create and re-create a specific event. With an enabler like that, every time *the event* happens Alice is reassured of the identity of Bob and at the same time Bob is reassured of the identity of Alice. Luckily such enablers exist in math since 1779 AD and new ones have been proposed since then[4].

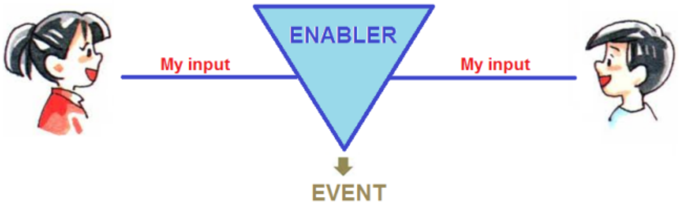

Figure 3

The important conclusion and the significant security advancement in the proposed new way is that two parties will be able to identify each other with absolute certainty solely by the occurrence of an event, without having to interrogate each other as we do today with shared secrets. For example, if a husband (party A) needs his wife (party B) in order to access their joint bank account, every time the husband gains access to the account (outcome of the event) he does not only benefit from online banking, but he also confirms that the party who collaborated with him and helped him access the account was his wife. So if in another instance, the husband for whatever reason has doubts whether a cyber-party who claims to be his wife is indeed his wife, he can find out by simply inviting the other party to help him access the bank account! If he gains access to the account, the other party is definitely his wife.

<u>Trust from mutual needs</u>

In practice, instead of interacting with one another, two cyber parties will interact independently with an enabler (process/system, non-human) to make a mutually beneficial event happen. This is how it would work:

> *"Two parties who are cyber inhabitants wish to trust each other every time they meet on the Web. These parties may be two humans, two systems, or a human and a system. Both parties agree on a process, system or mechanism (called the "enabler") that has two important properties: first it generates an outcome that both parties need, and second it generates it only when both parties participate. Participation can be different for each party and it can either be online or offline.*

[4] Math is out of the scope of this paper. If you are interested please contact the author.

> *Also a party's participation may vary from a simple approval, to providing missing info, all the way to providing missing part(s) of a process/system. The resulting outcome, or event, may appear the same or different to each party and it may occur online or offline. Regardless, the outcome of the event must be to the total trust-satisfaction of each party."*

Let's look at an example. Say two parties, you (party A) and a bank (party B), agree on an enabler that will control all VISA cards you obtain from this bank. Both you and the bank agree that the enabler, and only the enabler, will be able to enable and disable your VISA (the event). This means that you will not be able to control your VISA card; the bank will not be able to control your VISA card; it will be you and the bank together who will control your VISA card via the enabler. When the bank issues a new VISA card to you, both you and the bank will need to interact with the enabler to complete the issuance process and activate the card. The result of the issuance process will be a valid, activated but disabled VISA card. From then on, whenever you wish to make a payment with your VISA card, both you and the bank will need to cooperate with the enabler to temporarily enable your VISA info, use it to complete the transaction, and disable it again. You can see the mutual needs in action: both you and the bank need the VISA card info to complete a payment transaction, but neither of you can have it unless both of you cooperate with the enabler. This way, you cannot abuse your VISA, the bank cannot abuse your VISA and most importantly, hackers and ill-willed merchants cannot abuse your VISA.

This example demonstrates how two parties that do not necessarily trust each other, can trust payment transactions made by a VISA card because they need each other for every transaction. The same example also illustrates how two parties (card owner and bank) can identify each other via the enablement of VISA info, an independent event, which can only occur when both parties collaborate willingly.

We named identities which are based on mutual needs between parties and confirmed by independent events, *Eventities*. Next we discuss a transition from online identities to Eventities.

We live in two worlds

We live double lives. We live in the natural world and we are also inhabitants of the digital world. In the natural world, we exist as a single and unique physical entity and we are known to others by descriptions of ourselves called identities (IDs). Such IDs include passports, driver's licenses, Medicare cards, etc. In the digital world, we exist as digital descriptions of ourselves commonly known as profiles (digital IDs). As we go about our daily lives in both the physical world and the digital world, we use our IDs to interact with each other and we also use our IDs to interact with inanimate objects and services as well. Complementally, digital objects and services such as cars, phones, tablets, game consoles, Websites, clouds, Netflix etc., also have IDs and they use them to interact with us, and to interact among themselves. Every interaction in the natural world and every interaction in the digital world begins by parties using their IDs to recognize each other.

But what exactly is an ID and why do we have more than one?

Traditionally, an identity (ID) is a set of characteristics and attributes of a person, object or system in the natural world, and in the digital world, an identity is a set of characteristics and attributes of a system, object or service. Thus, identities are nothing more than descriptions of physical or digital entities. But why are they many? Why don't we have a single universal description (ID) per entity? The answer is because we cannot guarantee the completeness of the list of skills, characteristics, attributes, and other aspects of an entity included in a universal identity. Actually this list is endless considering that an entity can find itself in myriads of physical or online situations. Trying to define a universal ID for an entity is like trying to predict its future. Since this is not possible and universal IDs are a utopia, we cut the endless list of descriptions of an entity into specialized sets and create a number of IDs per entity. Each set, each ID, has a limited scope defined by the two parties involved and the situation they are in.

For example, if the two parties are you and a hospital, you can have all the IDs in the world but if you don't have a valid Medicare ID you will not be treated. In another example, if the two parties are you and a traffic-policeman, none of your IDs can replace your driver's license. IDs are used to determine whether the entities they represent have a specific characteristic, possess a specific attribute, can perform a specific task, or have a specific skill required. Precisely because all IDs are contextual and intentionally leave some descriptions out, they are not, and they will never be universal. This is the reason why we have several IDs in our wallets and why we have a number of IDs online.

Luckily we do not need a universal ID. What we need is a multitude of IDs to cover the concerns of parties we encounter. In other words, IDs are a private matter between two parties that wish to interact in the physical world or online, and if they are satisfied, it is nobody's business. Who is to say that a Website is wrong when it is satisfied with my ID, and who is to say that I am wrong when I am satisfied with the ID presented to me by a Website? This dynamic view of IDs is scary at first and sounds untrustworthy. However this is something we do every day. When you need a plumber, a carpenter, or someone to fix your shoe, you need to see an ID with certifications, credentials, testimonials and descriptions of his skills, and it is perfectly acceptable to leave out his name, his family history, his home address, his medical record, his driver's license and many more of his other qualities. The only thing you care about is whether he can do the job. Context is the filter that dictates which aspects of an entity should be left out or be included in an identity set.

In summary, IDs are sets of (multimedia) info about an entity which are used to represent the entity in different situations with different parties. But how do parties authenticate identity-sets presented to them by other parties? And what is the link between information about an entity and the entity itself? Unfortunately, ID verification systems today do not check the authenticity of the content of IDs and have no direct way to link entities to identities. Remarkably, identity-sets are not authenticated before use. Instead, ID verification systems today automatically trust the identity of a party, i.e. they automatically trust info included in the party's identity set, if the party knows a shared secret (password) associated with it[5]. Clearly, there is no link between the knowledge of a shared secret and the authenticity of the content of a party's identity set.

Eventities solve that problem by forcing interested parties to first agree on the content of an identity-set and then disable it together. For example, a registered user (Party A) and a Website (Party B) agree to encrypt the user's account profile together (the event). However, the algorithm they used is a cryptographic algorithm that can en/decrypt the user's profile only if both the Website and the register user collaborate and contribute. Following that, the user's profile (account identity-set) may be decrypted in the future for a variety of reasons. Website may wish to identify and authenticate the register user before it logs him in; the register user may wish to identify and authenticate the Website before he starts using it; the Website may wish to identify and authenticate the register user before it lets him access the account, a service or a resource; the register user may require Website's consent before he can make a payment or take a collaborative action, etc. In every case both parties must agree to participate and contribute.

Using multi-party, mutual-need functions such as the cryptographic algorithm[6] in the example above to create Eventities has several advantages. First, at rest the user profile remains disabled, thus preventing everyone (hackers) from accessing the user's account. Second, the Website alone (inside personnel) cannot access the user's account. Third, the registered user alone cannot access

[5] See sister document titled "What is wrong with passwords?"

[6] En/decryption algorithms are one of several ways to en/disable a digital asset or service

the user’s account. Fourth, since both parties encrypted the content of the user’s profile together initially and since both parties are subsequently required in order to decrypt it, both parties are reassured of the authenticity of its content. Fifth, en/disablement of the user profile proves the Website’s intent because en/disablement cannot occur without its participation. Sixth, en/disablement of the user profile proves the registered user’s intent because en/disablement cannot occur without his participation. Seventh, access transaction logs are for the first time admissible in court of law and can irrefutably exonerate or implicate the registered user and/or the Website.

A digital identity that cannot be stolen

Hackers do not steal. They make a copy. Two reasons. They make a copy because it is the easiest and fastest thing to do with digital systems, and they make a copy because they do not want their victims to know that something is missing. Therefore, if we want a digital identity that cannot be stolen, it must be a digital identity that cannot be copied. A digital identity that cannot be copied is a digital identity that is not data (data can always be copied). If a digital identity is not data then it must be function. And not just any function. It must be a function that allows two parties to identify each other by collaborating to make an event happen. It must be an Eventity.

Eventities are identities that cannot be stolen because they are not tangible. Eventities are unique ephemeral abilities of participating parties, active for a brief moment and then disappearing without leaving a trace.