Bootstrapping Adoption of the Pico Password Replacement System (Transcript of Discussion)

Frank Stajano(✉)

University of Cambridge, Cambridge, UK
frank.stajano@cl.cam.ac.uk

I spoke about Pico in 2011 at this workshop. This is not about how to build it, but how to *bootstrap adoption* of the Pico password replacement system. All the researchers in the Pico team who have contributed to this work, whose names are on this opening slide, are here today. In 2011 Pico was just a dream and I'm very glad now to have been able to recruit people that made this into something real that we want people to actually use.

Eliminating passwords has now become even more of an imperative because passwords are really impossible to deal with, at least in the way that we present them to a normal human being:

> "Pick something you can't remember, then don't write it down",

as Mikko Hypponen once overheard and tweeted[1]. If you get hacked after not actually following these impossible instructions then you get the blame. Is this a fair thing? I don't think it is. We technologists should offer people something they can actually do. Passwords are pathetic from the viewpoint of usability, they are also not very good in terms of security, and so we need to improve on that. So this is why I came up with Pico.

For those of you who were not here in 2011, we have made a little movie to show you what this is about.

Readers of these proceedings are encouraged to watch the 7-min video on our web site, http://pico.cl.cam.ac.uk, *because we are not reproducing a transcript of the movie here.*

This is an envisionment video, of course: the device shown is not the real Pico but merely Graeme's highlighter marker! We showed you how we envisage Pico will work when we finally build it.

However, we've got a problem: compatibility. I said we wanted to be bold and explore the whole solution space. Imagine a Venn diagram with three nested sets. The innermost one is passwords. The one that encloses it is the set of password-replacement solutions that are still *compatible* with passwords, so you have some hope of actually working with what exists today. However, around it, there is a much bigger set: a solution space of systems that might replace passwords but are *incompatible* with them. Not many researchers go there, but I wanted to make sure we are not missing out on something good that only exists over there.

Pico doesn't constrain itself to the inner sets. This gives us great intellectual freedom but of course it also means that Pico is incompatible with what currently

[1] https://twitter.com/mikko/status/102984155809333248

B. Christianson et al. (Eds.): Security Protocols 2014, LNCS 8809, pp. 187–196, 2014.
DOI: 10.1007/978-3-319-12400-1_18

works today. When you have a Pico—that's great!—what do you want to do with it? You want the freedom of not having to type a password every time you log into Google, or into Amazon, or into Expedia, or into whatever else you use—but you can't, because, at least today, these sites are not Pico-compatible! They don't even know that Pico exists.

We have essentially a "vicious circle of adoption": this is our big problem. Nobody has a Pico because we haven't built it yet, and this means that websites won't support Pico because none of their users have one. No website accepts the Pico alongside passwords, and therefore this means that users have no incentive to buy a Pico, because they can't use it to log into any site that they might actually want to visit. As a consequence, no users have a Pico, and we have this vicious circle. Today's presentation is about the work we have been doing to *break* this vicious circle.

We have developed several pieces of software in order to break this circle in three different places. So we're going to talk about three pieces of software. The first is the Pico Lens, a plugin for your browser that displays the websites as if through a magic lens that makes them appear as if they supported Pico even though they don't. So it's no longer the case that "no websites accept Pico alongside passwords" because, if you look at them through the Lens, then suddenly they *do* accept Pico. The second is the Pico Prover App for smartphone, which is essentially a version of the Pico gadget as an app. This *(tongue in cheek)* is evil! I said from the start in 2011 that that's not the way it should be done, because your smartphone contains software you download from strange places thinking it's as good as Angry Birds, and instead it's probably malware, and you don't want to have all your login credentials sitting next to that malware, so from the security viewpoint you are in a state of sin if you make a Pico app. But, on the other hand, if you are trying to break the vicious circle, the advantage of making a Pico app on the phone is that the user doesn't have to buy a Pico because they already have the smartphone; perhaps even more importantly, the phone is something they already carry: we don't have to persuade them to bring along another gadget. So, this way, they get a Pico-equivalent device in their hands, relatively painlessly. The third piece of software we developed is the Pico Verifier plugin for Wordpress-based websites. This means that websites don't need to invest in a major software development effort to support Pico because, so long as they are running Wordpress (and 60 % of the websites that are run on a content management system are Wordpress-based), then they can just adopt our Wordpress plugin, and they will automatically support Pico without effort.

So let's have a look at these one by one. The Pico Lens is, as I said, a browser plugin such that, if you visit a website that doesn't know anything about Pico, when you look at it through the lens it appears as Pico enabled, and you can interact with it with your Pico and it works.

Joseph Bonneau: Is it technically a plugin or an extension? A plugin is like Flash, it can read your hard drive.

Reply: The Firefox APIs have changed recently: technically these days it's called an add-on. The API that we wrote for originally, when we starting doing the

Lens, has also changed and has been deprecated. Chris can give you all the details offline if you wish. Anyway, to explain the Pico Lens and also to demonstrate the Pico App on the phone, we prepared another small video.

The video, not transcribed here, showed the then-current prototype of the Pico App on the Android smartphone logging into two non-Pico-aware sites, Gmail and LinkedIn, through the Pico Lens in the Firefox browser on the laptop. It also demonstrated the initial pairing phase between a new site and the Pico.

Let me point out the crucial difference between the previous video and this one. The previous video, which we did in December 2013, was an envisionment of what we wanted to do. This other video, from a few days ago... OK, it's completely flawed, there's a bunch of stuff that we want to change, but this actually *works*, we built some software to implement this. This is tremendously exciting! We can break the vicious circle by releasing this app and this add-on and letting people use the Pico system even *before* the big websites start to support it. With this, we are independent of everybody else having said yes to Pico. We can all start using it as soon as possible.

In developing this browser plugin (I'll keep calling it a plugin even though that may no longer be the most technically appropriate term) we had to face some interesting design decisions. We imagined: let's make the plugin in such a way that the site seen through the lens is as similar as possible to the real site. To do that, we would have had to basically use the same protocols as the Pico device would use with a real Pico-compatible website. Normally the real website offers its public key in a certificate, which is in this QR code; then, from that, a channel is established between the site and the Pico, and so on. Now, if the Lens plugin were to generate a public key for the site on its own (a key which, of course, the real site doesn't know about), and offer that to the Pico device when scanned, then who would have the secret key for that public key? Not the real site, because Google or LinkedIn or Expedia doesn't know that we are doing Pico. So the key would have to reside in the plugin itself, but that's dumb: what happens if you visit the same web site through another computer (with its own Pico-Lens-equipped browser)? You'd get a different public key for this website. So this doesn't work. Then we thought that maybe we can have some back-end, say http://mypico.org, that holds the master copy of all these fake key pairs generated by the plugins, and then all your plugins on various computers could synchronize through that back-end. But we didn't really like having this trusted third party somewhere in the network and having to depend on http://mypico.org being up 99.999 % of the time. We'll never have the same number of nines as the big websites like Google and it would be painfully embarrassing if your authentication (to an unrelated place) failed because our site was not up at the time.

So, how does this work now? In a way that is rather more convoluted. In fact, we do something different when the Pico talks to the Lens plugin as opposed to a real Pico-enabled website, so we have to maintain different strategies for the two cases. The Pico device has a direct connection to the website; it may be tunnelled through the terminal, but the Pico talks to the website over an

end-to-end-encrypted SSL channel, and it does its own login, by sending the password; then it obtains a cookie, which it then transfers to the plugin in the web browser. The Pico transfers this credential just for that session, so it doesn't actually even tell your password to your browser, unlike what happens with a password manager in the browser, where the password manager knows your password and sends it to the website you visited. In this case the password is in your Pico device, and the Pico device gives the password directly to the website at the other end, receives a cookie and passes this cookie as a kind of session credential to your browser, which then logs in. And then, if the browser is compromised, you've only lost that cookie, as opposed to having lost the password. So, in some sense, this is slightly more secure than using the password manager in your browser, even though conceptually it is very similar.

Virgil Gligor: What if you lose the Pico (or phone) that has the initial password?

Reply: You have two concerns: one is confidentiality and the other is availability. As far as confidentiality goes, the device is locked. The Pico hardware device is designed to have all its storage encrypted; otherwise, with the Pico app, you rely on whatever protection your model of phone gives to all of the private data it stores locally. For the availability aspect, as shown in the first video, you have a backup: in the case of the hardware version of Pico, every time you recharge it, it gets backed up automatically. In the case of the software version we have something equivalent: whenever it learns a password for a new site, it automatically does a backup to the cloud, so there's a forcing function that doesn't depend on you taking backups personally. And then you can always just buy a new phone, reinstall the Pico app and restore your credentials on there.

Virgil Gligor: Do you mean the passwords?

Reply: Yes: in the case of non-Pico-enabled sites seen through the Pico Lens, your credentials would be your passwords. (Otherwise, for proper Pico-enabled sites, they'd be your private keys.) It's a similar situation if you have on your laptop a password manager storing all your passwords, and you lose your laptop. You can restore from the backup after buying another laptop. Hopefully your laptop had full disk encryption so the guy who stole it cannot access all your passwords. The same situation, really.

Alastair Beresford: You might be able to sell a USB charger that actually also does enough actual stuff, over either ADB (Android Debug Bridge) or perhaps just the protocol for sharing data from USB storage, to do the backup bit.

Reply: Yes. The interesting thing about this is that the original design that is already in the 2011 paper said that whenever you recharge the Pico, it takes a backup; and I thought I was very clever because nobody ever does "backup", but everybody does "recharge", so I got the backups for free. But now, if Pico becomes as an app on the phone, then we cannot guarantee that people will recharge the phone through that gadget you're mentioning, even if we build it,

so it doesn't have the same "forcing function"[2] property as the original solution. The original design is all very nice but, if we are also distributing a Pico app, we have to find some other way to make sure that the backup is taken, because people won't necessarily be using our docking station or your USB charger for the recharge—they might just charge the phone with any random USB cable, because that's all they have to hand. What we are doing for the Pico App is that, anytime the Pico goes online to talk with a site, once it has done its bit, since it has network connectivity it can also do a cloud-based backup. We've been talking to Ben Laurie about various things and he suggested that we use his Nigori[3] for doing the backing up that the docking station would do. This is a possibility and we'll consider it.

OK, so that was the Pico Lens. In the second video, you've also seen the Pico Prover App on the smartphone. The prototype just recently started working and what we need to do now is to make it practical to use. We claim that Pico will be easier to use than passwords; at the moment it isn't, and it's a bit slow, but the technical mechanics of it are there, and it's exciting to login with it! I repeat, this second video is not Hollywood, these were the real Google website and the real LinkedIn website. We have to deal with the same issues as the normal password managers in the browser: you have to write ad-hoc code for many special cases because there isn't really an API for logging into a website by machine. You have to catch many variations in web page design, things that people do in different ways, and hope that you cover enough cases. Even the in-browser password managers who have been at it for years sometimes get it wrong, so of course we are getting it wrong a lot more; but these are the real websites, we are not faking the back-end.

Ultimately we want Pico to be a single-purpose trustworthy hardware device: it does nothing else, it doesn't have other software running on it that can become malware, you have no reason (and no way) to install other stuff of dubious provenance, and there is hardware security to protect storage and to provide a trusted path between the user and the display and the application. However it is very hard to make people buy a new gadget and it is even harder to make people *carry* a new gadget, even if the gadget were free. It's kind of miraculous that people have been converted to carrying a mobile phone: almost everybody carries a mobile phone nowadays. But it's a really tough uphill struggle to make people carry one more thing, so it's much easier to get this Trojan horse on a piece of software, on something that they already carry. (Of course I am saying things in jest: my marketing department, if I had one, would tell me that I should never use such tainted language to describe our wonderful software! The Pico App is in no way a Trojan horse, it's a marvellous gift that will make people wonder how they could ever live without it before.)

We suggest you start by using the Pico App on your smartphone only with sites that don't require extreme security. Most of the websites that you visit and type passwords into don't really need super-high security: they're not all

[2] In the interaction-design sense of a behaviour-shaping constraint.
[3] http://www.links.org/files/nigori-overview.pdf

the same as your online banking and you can get away with some low-grade password. Well, you could easily use Pico for these sites while you are not yet sure whether you trust Pico or not.

Our third piece of software is the Pico Verifier backend for Wordpress sites. We chose Wordpress because it's used by 20 % of all sites and, as I said, by 60 % of the sites that are based on a content management system. In this slide you see what a Wordpress blog looks like when you enable the Pico plugin: next to the username and password prompt that you're familiar with, you also get a QR code. If you scan it (assuming you already have an association with that blog) then you get in. How do you make the association if you don't have one? You just log in with your normal password account and, on the settings page in the blog, you will get another QR code which, if you acquire it, will form a pairing between your Pico and your account in the blog. From then on, you can log in with Pico.

What is the socio-economic context in which we are trying to get Pico deployed? Remember the famous saying that "nobody ever got fired for buying IBM"; well, it's similar here: nobody ever got fired for using passwords to authenticate users. We have an incumbent in this field (passwords) that is going to be pretty difficult to displace, because it's what everybody else does, and there's no reason to do things differently, and there's a very big "activation energy" barrier to overcome. As I said, it's also very hard to persuade normal human beings to carry another gadget. When talking to banks and insurance companies we also discovered another barrier to overcome: they worry that they ought to be able to sue someone if the system doesn't work or gets hacked. If someone *sells* them a password-replacement solution, then they can sue them if it doesn't work. But if Pico is an open system they wonder: Where is my recourse? Who bears the liability? If this is something anybody can build and implement, how will we manage the risk of not being sure if this is going to work?

There's another potential issue, which is something we have seen in the history of our industry at the end of the last millennium. As you may remember, Sun came up with the Java programming language—an open, standard language that anybody could use or implement. Microsoft in response implemented a version of Java that worked better with Microsoft products: they did that by introducing some additional instruction that only the Microsoft Java Virtual Machine implemented. Sun got really upset because Microsoft's incompatible version threatened their "write once, run anywhere" vision. And Microsoft was so big, with their widely deployed base of Windows, that, if they changed the JVM, then for most people it was going to be just the Microsoft Java that worked well, not the real Java. Sun sued them in 1997 for 35 M$, and they settled out of court after several years. We don't want something like that to threaten the universality of Pico, either: this would be the dystopian scenario of one big web company making their own incompatible version of Pico so that they "own" it and control it, and this bastardized Pico becomes the de facto standard just because they're big.

And then, another issue: where's the incentive for people who might manufacture the hardware Pico? There isn't a lot of money in a very low-margin product like this, so you only have an interest after you are sure that there is a market of billions of people who actually want a Pico.

I am very happy that I got this team of people who are making this dream a reality, and especially that we found a strategy that does not pollute our dream. We are not compromising our design by trying to be compatible with passwords as a prerequisite, and yet we can still allow early adopters to enjoy the Pico benefits.

What next? First of all, try the Pico app and the Pico Lens for yourself, as soon we release it, which we hope will be soon. If you are a user, tell us what it would take you to use Pico for authenticating. If you are a representative of a company that uses passwords to authenticate its millions of users, then what would it take *you* to adopt Pico? This slide shows a paper design, by some of our undergraduates, of what the Pico hardware might look like. This final slide lists all the people who have worked on Pico so far, including the members of the team, who are here, and the past and present project students, one of whom one is here as well. Thank you very much.

Ross Anderson: The last time I bought a £10 mobile phone, it was being subsidised from their anticipated future sales of minutes, because it was locked in. And similarly, at an RSA conference three years ago (I think it was), I got hold of one of these VeriSign Identity Protection devices, which is a one time authentication token that they give away free. Their business model is that they charge 10 cents or 20 cents per authentication to the websites which use it. Their clients are typically people like stockbrokers, who place a significant value of risk, but are not big enough in terms of number of users to roll out their own custom authentication infrastructure.

Reply: It is true that in the past the £10 phones were subsidised by network operators selling you minutes. Nowadays, however, this is a £10 phone that I bought in Tesco, SIM-unlocked. I just paid £10 for it and nobody tied me to any contract. So, once the market is big enough, there is obviously enough money to be made just by selling you the £10 phone. I want to figure out how can we get to that stage with Pico.

Bruce Christianson: You start with drug dealers!

Rubin Xu: For the Pico Lens, presumably for some web sites you still need the users to remember their password to periodically refresh their authentication cookies?

Reply: Well, not quite. The Pico does it for them. Every time you log in via the Lens, your Pico sends the password directly to the real website; it just doesn't send it to the browser.

Rubin Xu: I thought you said the Pico doesn't remember your passwords.

Reply: The full Pico protocol uses public key crypto and uses no passwords at all. However, when the Pico device (which in this case is the Pico App on your

phone) accesses a site through the Pico Lens, it has to use (and remember) your password instead, because the site itself doesn't know about the public keys of the true Pico protocol. The Pico just doesn't tell your browser the password; instead, it talks to the website directly and tells the website the password. When the website replies with an authentication cookie, the Pico gives the authentication cookie to your browser, so that the browser can actually have the session.

Unidentified Audience Member: You pass along this cookie through some different channel?

Reply: Yes. *(Draws a diagram on the flip chart and refers to it.)* This is the website, this is your terminal that contains your browser, this is your Pico device (which in the video was on your phone), this is the internet, and the terminal connects through the internet to the website, and the Pico talks to the terminal with some short-range radio channel. The Pico can tunnel through the terminal or it could, by connecting to the internet on its own, talk to the website directly.

We already have to have this local connection from Pico to browser for reasons that will become clearer in Max's talk, so through this connection we tunnel a direct connection from Pico to website, through which the Pico does a login as if this were a web browser on its own. And then, when it gets a cookie, it gives it to the browser, which then uses it in a connection that is somehow faked, because it was established by the Pico, but which the browser pretends is its own.

Virgil Gligor: Couldn't this be subject to a relay attack?

Reply: That's going to be the subject of the next talk!

Bruce Christianson: Do you have a plan to break the new vicious circle which you've got now, where people won't buy a Pico because they've already got a mobile app that does it, and websites won't implement Pico because their users can just use the Pico Lens?

Reply: The fact that people might just be content with a mobile phone is a matter of where your personal security trade-off is. If you're sufficiently paranoid that you want to have something more secure than a smartphone app, then, for what we hope will eventually be the £10 of today's low-end mobile phone, you can buy yourself a gadget that's stronger security-wise, besides being easier to use because it's dedicated rather than general-purpose. Concerning web sites, if you use the Pico Lens you're getting some of the benefits of Pico, but not all of them, especially on the security side. If you're a website and you are fed up with all the security problems of passwords and having to deal with all the incidents, then you really benefit, in security, if your customers switch to using Pico, especially if they already have a Pico for other reasons anyway.

Joseph Bonneau: This is a follow-up on what Bruce is saying. It seems like one of the security benefits, to a website, of supporting Pico, is that, for example, it would prevent phishing. But that's only a benefit if users were sometimes willing to type the password and still remember it. But, if they're doing that, then it implies there is some usability benefit why they would occasionally want to type the password themselves at other terminals or whatever.

Reply: I am not sure I fully understand the question. The sort of patch that we are deploying now (the Pico Lens) is something that you, as a user, adopt because you get many of the usability benefits of Pico, but not all the security benefits.

Joseph Bonneau: What I'm saying is: take any security benefits from switching from the phone-based sort-of-fake Pico without the website support to a full Pico, one of which would be phishing resistance. Aren't these benefits already granted even with the fake Pico Lens, assuming that users no longer type the password?

Reply: Not quite. The Pico Lens still relies on the website accepting normal replayable static passwords, which is what it sends behind the scenes. So most of the inherent security flaws related to the use of replayable static secrets (that can be overheard and then replayed, or leaked at the far end and so forth) are still there. Even if the user never types a password, the website could still get hacked and lose the (hopefully hashed and salted) password file, for example.

Rafi Yahalom: I was wondering: for the smartphone manufacturers, what would they have to implement at the infrastructure level to have the Pico application be enough? Is there any chance of that happening?

Reply: The smartphone manufacturers are already doing things such as using ARM's TrustZone for separating apps within the phone. The thing that is still missing is a trusted path from the user to the app. Right now, you can't easily tell if another app puts up a screen that looks like another's. Even though the malicious app cannot access the data of your good Pico app, the malicious fake app can still look like the Pico and trick you into doing things, and that's something that's missing.

Virgil Gligor: So you need a Wimp!

Max Spencer: Bruce mentioned a "new vicious circle" where there's no incentive for the service providers to adopt the full Pico protocol: well, actually there is! The way the Pico Lens works is that it takes the username and password that you previously typed and tries the login form again; it's not like a bank login where I have to have my little separate token: there, the Lens clearly wouldn't work. So, if there is now this positive spiral where people have Picos and use them for all the services that do support them, then there's pressure on the more security-conscious services (like banks) to adopt the full protocol so that they become Pico-compatible, given that people are now in the trend of wanting to use it.

Partha Das Chowdhury: For the Pico mobile, does the phone company need to implement any Picosiblings?

Reply: You raise a good point. At this stage, the smartphone prototype we are deploying doesn't have all the locking mechanisms that we described in the original paper. We are just implementing the interaction between the Pico device and the remote site, while relying on whatever the phone does to lock itself. This could be the usual PIN, the usual squiggle, the fingerprint recognition if you have the latest iPhone that costs £500 and so forth. But that's of course not the full

design. Would the phone app also implement the Picosiblings? It depends on how much we separate the two parts of Pico (interaction with the verifier vs locking of the Pico token). If the phone did the Pico-style locking, it would also have to talk, locally, to your Picosiblings; maybe not all phones will have hardware that can do that.

Partha Das Chowdhury: In the actual implementation, what's the right number of Picosiblings you will have?

Reply: For now we can only guess. We'll have to determine the most appropriate number through user trials once we build prototypes of that hardware.