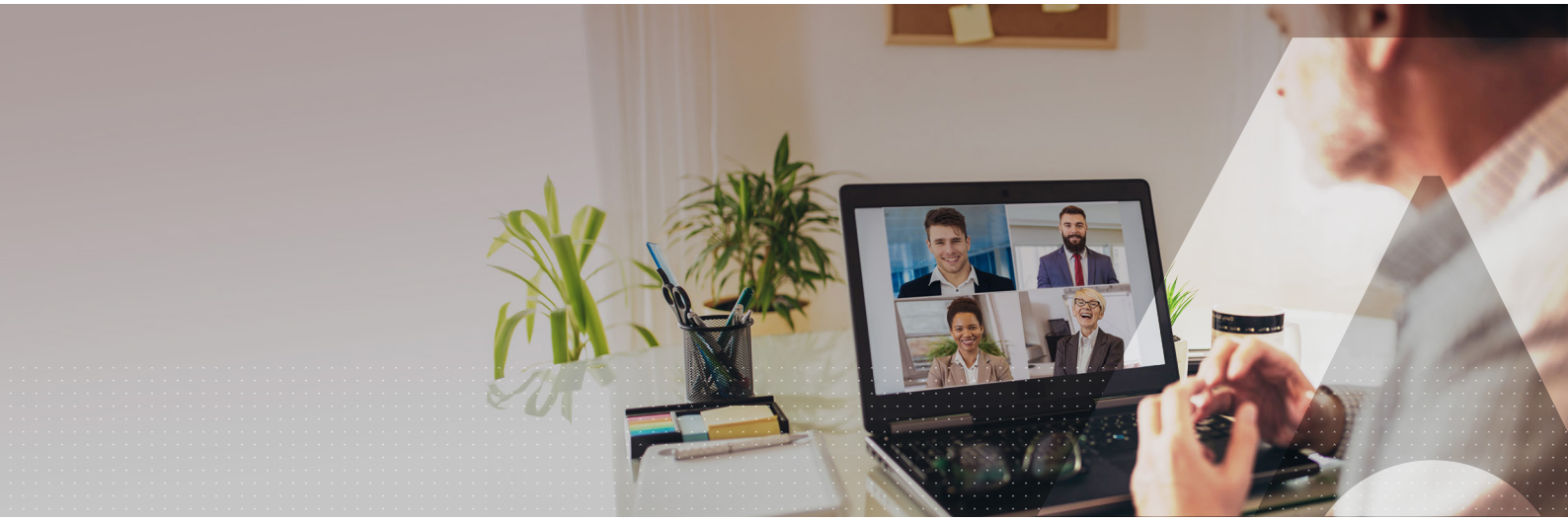
cpl.thalesgroup.com

THALES
Building a future we can all trust

Trust Anchor in Modern Authentication



White Paper

Contents

Introduction

You might consider this a boring subject, but it may make you sleep better.

The increasing proliferation of authentication methods makes it necessary to establish a trust anchor for the various methodologies. It is becoming more and more dangerous to continue using various authentication methods without checking the trust anchor occasionally. Of course you could always use the trust anchor for authentication, but that may not be convenient for users and some applications may not be able to address it. In everyday life it is not always necessary to recur to the highest level of trust. For normal processes intermediate and more convenient solutions can be good enough. You don't always wave your passport when entering a building or buying a burger, in which case you might rather wave your healthcare card or some cash. We have to keep in mind that malicious intruders may have substantial resources and IT competence when deploying their attacks. Large corporations or institutions are high value targets and need to protect themselves against highly professional attacks.

In normal life, our passports serve as a trust anchor, what can we use as an equivalent in the IT environment and what requirements or criteria should it fulfill?

An authentication can be seen as a kind of cryptographic signature transaction or procedure where one party asks for proof of identity with the second party providing it. It follows that we can relate to the requirements for digital signature for those requirements.

Trusted issuance

The first requirement is to have an issuance process where you can make absolutely sure that the right person gets the trust anchor. If that process is weak nothing that follows can be considered secure.

That is the reason why you have to present yourself in person at a government institution to get a passport. Actually it is the ultimate biometric authentication to meet an approved person to apply for a passport.

We have seen a lot of criticism in Germany for the health care card system because Chaos Computer Club was able to capture other users' health care cards, simply by using the manual process, writing to the health insurance pretending to have a user address change and request a new card. Not only was there no need for sophisticated IT knowledge. The best cryptography in the world is useless if the issuance process is weak.

So, how do users get a particular authentication token? You may be able to download an app and register it with your company, for example. How does the company know it's you though? Maybe you do it by authenticating first with another previously approved method? OK, that leads to some kind of regress. How did you get that one? Maybe by an OTP sent to you via SMS or Email? How did the sending server know where to send it? Probably because you have entered the phone number or email address, right? How was that authenticated? Maybe by looking at the Active Directory entry where you are authenticated with your password? So in essence the security strengths of your latest app is that of a password. Microsoft Active Directory needs a password or a certificate. Passwords are easy, hence still in use very

often, certificates not so much. How did you get your Active Directory entry in the first place? Maybe by an administrator typing it in. How does he know you? How does the company know he hasn't entered any number of other, maybe faked identities? It has happened before.

Of course there are different ways how to design a trustworthy issuance process. You could take the traditional approach, go to a helpdesk, present an ID and get a pre-staged envelope with your token handed to you by a security office who compares your picture on the ID with the one in his database and your face. After that he asks you to provide your signature, which he then compares with a previously stored picture and then hands over the envelope.

There are other, easier ways, such as multiple approvals by trusted people via their trust anchors. Many systems offer workflows designed to guarantee a secure issuance of a trust anchor.

Exclusive control by user at time of usage

The best trust anchor isn't worth very much if it's not in the exclusive control of the user at the time of presentation. That would be the problem with most single factor authentications. If for example the possession of a token would be enough to enter the system, you would never be sure who's entering it. It's used in buildings all the time. Stolen passwords are probably still the most frequently used example in the IT world. Another example is a token shared by several users.

The same would be a biometric image taken with your webcam by a spyware. The spyware takes your picture or short video clip for liveliness detection, combines it with a different user name and uses it for example to establish an account on a criminal website.

The biggest challenge here is the widespread usage of Identity Federation and single Sign On (SSO). The problem of regression of trust appears in all those scenarios. If a SSO gate is the only point of authentication, it not only becomes a single point of failure for service availability, it also becomes the ideal target for cyberattacks.

The user control should be checked occasionally to be able to detect misused credentials.

Independence of trust system

A trust anchor should not depend for its integrity on the very same system it's supposed to authenticate to.

With a passport the solution is self-evident. Of course your passport has no connection to the system of a border control officer checking it.

In IT systems however it's less evident. Could you for example consider your Microsoft Active Directory (AD) as a trust anchor? The answer is no, because the authentication is usually done against the AD. This means that it cannot be the trust anchor because a compromised AD entry would never be able to resist the very same compromised authentication request.

So what about the database of your authentication server? It has registered all the users, their token ID's, PKI keys and passwords. It is independent from the AD or application asking for the authentication. So, could it be a trust anchor? The answer is no as well. Admittedly it would be better than the AD as it's an independent system from the application requiring the authentication, but it is not under the sole control of the individual user.

There may be exceptions though. For one thing the entries could be private PKI keys. In that case the user could be in control if he is the only one knowing the PIN code.

Resistance against forgery

Fake passports are the classical example of invalid trust anchors. In a way the IT world has some advantages over the analogue world of passports. We have actual possibilities to make a forgery virtually impossible.

A PKI token can set up private keys uniquely without any possibility to duplicate it. On the other hand the IT world has to cope with problems that passports never face. An IT trust anchor has to work in completely virtual environments were physical passports have no meaning.

A special consideration is the "break glass in case of emergency" workflows when users have lost their authentication token. Very often these workflows are handled as exceptions with weaker security requirements than the standard process, which of course makes them the ideal attack vector. In those cases a recurrence to the trust anchor is highly recommendable.

What could be a good trust anchor?

So, what could be a good trust anchor? First of all we should keep in mind that this is not only a technology topic. It is more a process issue. The first requirement is a trusted process. By definition there is no "one fits all" solution. Actually nearly anything could be a trust anchor as long as the process to get one is secure enough. Not much sophistication is required. A signed piece of paper could be one in the real world. In the IT world it's less obvious though.

Operational problems

If you are in a closed environment i.e. a limited number of well-known people to access a predefined on-premises IT system, the best way would be the traditional handing out of a physical token to each person by an authorised security person, checking an HR database. It's easy to setup, it's intuitive and it's extremely difficult to attack from the Internet.

However it has practical limits when you have many people in different regions. The sheer number of cases may make it very expensive to execute. It has a long response time in case you need replacements for lost devices. In case people are working in remote offices it would be logistically difficult to supply physical tokens and control who actually signs for receiving them. The problem multiplies with the number of uninformed users, for example in consumer systems. On top of that people tend to find "creative" workarounds if things are too cumbersome, and almost all of those are security nightmares.

Online systems

The main benefit of an online system is the ease of scalability when compared to the physical logistics problem. However, if you use an online system as the trust anchor, how do you know you can trust that online system itself?

One possibility would be a "Know Your Customer" approach. You ask the user to present his passport, i.e. the old analogue world trust anchor, hold it to the screen and perform a face recognition based on the passport photo. It leaves the old problem of fake passports.

Another way would be to look at a pre-registered user's past behaviour, maybe in a federated system. The more confirmation from another federated source you can confirm, the more unlikely it is that someone could fake it. However it leaves the problem of trust to the federated sources over which you may or may not have any control. There have been trials with Blockchain based systems, following that logic.

It would also be comparatively easy for a patient attacker to build up a legend of persons to be used as identities at some time in the future. You may ask any fan of espionage thrillers for details.

The most frequently used version however is a mix of IT system permissions granted by one or more authorised users, assuming those would know the person requiring access, AND have valid trust anchors themselves.

There is probably not one best solution. All have their pro's and con's and need to be considered with financial limitations.

Trust anchor products

Let's consider some products that Thales and other companies offer.

PKI Smartcards

This is arguably the closest version of an IT passport. It has the benefit of following the traditional passport workflows that people are accustomed to. It may even carry the user's picture. Let's see how it compares to our four criteria:

1. Issuance
This is almost natural, you can hand it out like a real world passport. Being a physical item an internet attacker has no way of stealing it.

2. User control
Assuming the user keeps his PIN code confidential, it's ideal. Since the keys on those cards cannot be copied, the user can easily stay in control.

3. System independence
By definition smartcards cannot be compromised by the requesting system. Every user has his own card and his own PC. Hence there is no real single point of failure. It's a diversified system. Disabling a large number of users would only be possible by taking down system components like domain controllers or network hubs. The smartcard admin system is not involved in the authentication process itself.

4. Forgery
A smartcard can create a PKI pair and publish the public key. Since it's practically impossible to recreate the key pair, those cards cannot be copied. Well, at least as long as you don't have a working quantum computer, but that's another story.

FIDO token

A physical FIDO token is very similar to the PKI card. The form factor of an USB key or a card makes no difference as far as the trust anchor usage is concerned.

The difference is the root of trust established in an X509 certificate, issued from a trusted PKI system. The FIDO token can create a key pair but it lacks the root of trust. This has to be replaced by an identity provider (IDP) solution, which puts the burden of the secure issuance to the IDP system.

Unfortunately the IDP system itself will need a trust anchor. So, using a FIDO token for that purpose may not come as natural as with the PKI card.

Two solutions come to mind when looking at this:
a) There could be a token like the Thales IDPrime 3940 card. This card has both function, the PKI applet and the FIDO applet. It could therefore use the PKI function for the trust anchor and the FIDO function for the easy access to cloud applications and other FIDO enabled system.

b) Create an issuance process for the physical token via a security officer controlling ID

Biometrics

In a way biometric methods are the most natural way of authentication. It proves the person directly. However its usage has its own technical, legal and organisational problems. In most cases people use device biometrics of mobile phones. Those are technically weak security solutions that do not lend themselves to a usage as a trust anchor.

Let's look at the criteria:

1. Issuance
This is comparatively easy as it checks the person directly and there is no need for key exchange etc. It's more about whether we use device biometrics or do we operate a central system checking the biometric credentials? In case of device biometrics the issuance is not under the control of the company. You don't know whose fingerprint is actually used on the phone and you have no independent comparison. In case of a central biometric system you have to deal with the legal aspects of GDPR regulations and the high requirements for data security. E.g. Databases with biometric user data without encryption secured by an hardware security module (HSM) should be considered a disgrace for any IT professional,- and they are an invitation for GDPR violation charges.

2. User control
This is the easy part. The user control is secured, as long as the biometric system works. There is the false recognition problem, but it may be controlled by using a strict version, risking more false rejections which would require a different workflow to re-establish. However since the trust anchor function is not used very frequently, that could be acceptable.

3. System independence
This can be fulfilled easily. The systems controlling biometrics are usually independent anyway.

4. Forgery
Here we get into the complicated part. Device biometrics is not a secure way of doing it. Users can change their device biometrics without anybody knowing it. The biometric can usually be circumvented on the devices by exhausting the retrial counters and going with the replacement workflow. The credential storage could be compromised. Weak implementations can be deceived. Non-working biometrics need work arounds anyway. Fingerprints sometimes don't work after washing hands with highly detergent soap, after sweating in the summer etc. A professional biometrics system for example face recognition with database comparison work better, but are quite expensive or invite GDPR related issues.

Mobile phones

It may be natural to think of mobile phone as trust anchors. Users typically take great care of their mobile phones, especially when bought with their own money. The challenge is to establish a trustworthy link with the company and keep them secure.

1. Issuance
The first step is easy. Users have one already, mostly bought themselves. Linking it to the company is less natural. Typically you will download an app from a public webstore, which in itself is of course no trusted process. That app would then be registered with the company usually with some kind of challenge or response mechanism. However since the download of the app is no trusted process you depend on the pre-registered mobile phone number and a verification by someone to make sure the right person is actually in possession of the phone. It can be done by using a different channel e.g. the company's secure email or an email authorisation OTP. Still you don't know who is behind the phone at that moment.

2. User's control
As long as there is a verification of an authentication via a knowledge factor, it's okay. Otherwise everybody who gets his hand on the phone would be let in. Even then it could be compromised by spyware. Just look at the discussion in Germany about the "Bundestrojaner".

3. System independence
Credentials of which phone to use could be managed by a system calling on phone numbers. However as long as applications are setup to always request authentication through an IDP gate, it's okay.

4. Forgery
Mobile phones are essentially computers with a permanent online function. They can be compromised with the same methodologies as regular computers. Spyware can be used to get hold of almost anything in the memory. User interfaces can be manipulated to get credentials etc. The protection by security software is by no means better than on regular PC's. As a trust anchor we would not recommend it.

Summary & Recommendation

What is the conclusion? We would recommend using a physical token like a card or USB token as trust anchor. They are the most secure version, easiest to setup and not very expensive. Trust anchors need not be used frequently but should be the basis of issuing other authentication possibilities. You should check them occasionally. Mobile phones, while okay for day to day operation should not be used as trust anchors because they could possibly be compromised by an internet attacker.

About Thales

The people you rely on to protect your privacy rely on Thales to protect their data. When it comes to data security, organizations are faced with an increasing number of decisive moments. Whether the moment is building an encryption strategy, moving to the cloud, or meeting compliance mandates, you can rely on Thales to secure your digital transformation.

Decisive technology for decisive moments.

THALES

Building a future we can all trust

Contact us

For all office locations and contact information,
please visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <