What if I lose my phone?

And how do I mitigate such disasters before they happen?

Daniel Hardman
Chief Architect, Evernym
Secretary, Sovrin Foundation Technical Governance Board
March 2019

sovrin
identity for all

sovrin

identity for all

Pick Your Disaster

Self-sovereign identity (SSI) has to be robust. It should be hard for accidents or malice to wrest control of an identity away from its rightful owner--and it should be easy to fix problems if they do arise.

We can test robustness with questions like this:

> If I'm managing my self-sovereign identity with an app on my phone, and I accidentally leave my phone on a park bench, can a malicious person take over my identity? If so, how do I get it back?

There are many variations on this question. Instead of a lost phone, the disaster might involve a hacked server, ransomware, lost keys, or a bandaged finger that can't provide a fingerprint. However, prevention and recovery answers are similar across scenarios.

First Answer

The technology that Sovrin[1] brings to bear on this problem is complex and nuanced. We could list its ingredients here--but such a list would feel long and mysterious without a lot of explanation. Besides, like a recipe, the result is more than just mixed parts. So instead, let's begin with a narrative about the experience of a Sovrin user named Alice, who's just discovered that her phone was stolen from a park bench. For now, let's assume that the phone has only weak protections--maybe a PIN, but no encrypted file system, secure enclave, or strong biometrics. This is not Sovrin best practice, of course, but it is all too believable, and it lets us explore some unique answers.

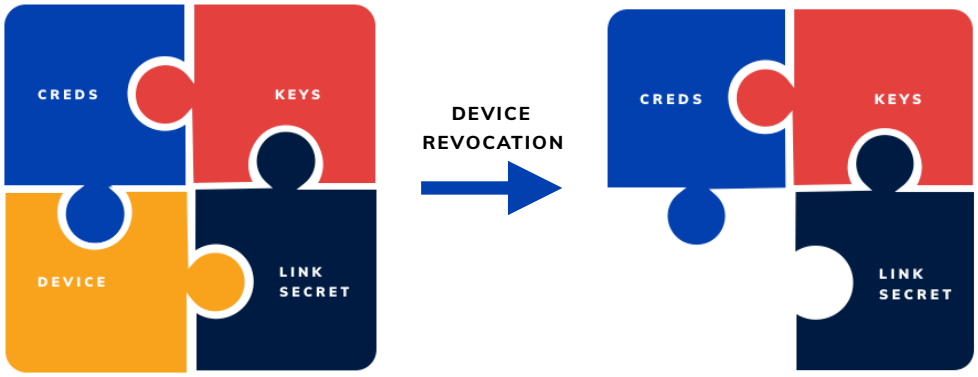
STEP 1: REVOKE THE DEVICE'S AUTHORIZATION TO USE CREDENTIALS

The first thing Alice should do when she discovers the problem is to cancel the permission of the lost phone to represent her in digital interactions. This "device revocation" will drastically limit the mischief that a thief can do.

[1] Most of the discussion here applies to Hyperledger Indy as well as to Sovrin, since Sovrin relies heavily on features in Indy's codebase. However, Sovrin is mindful and choosy about how Indy features should be used, where Indy is more flexible. Also, challenges to our answer are likely to touch on the governance framework that is Sovrin's unique contribution. Thus, the discussion assumes Sovrin.

Alice does this by using any other device(s)[2] she controls[3] to write a transaction to the public ledger. Recording such a transaction can be done in seconds, and it takes effect very quickly, worldwide. It does not require any connection to the stolen or hacked device, or to any credential issuers, or to all of Alice's connections--and Alice can do it in a privacy-respecting way.

Device revocation works because impersonating Alice in Sovrin requires more than just her credentials and secrets--*it also requires a device she has authorized*. Credentials are only useful if they're used from such a device; if Malfoy (the thief) moves them elsewhere he accomplishes nothing. Thus, device revocation doesn't change the status of Alice's credentials; she can continue to use them wherever she likes. It simply makes the compromised device useless as a representative of Alice.



Device revocation keeps a hacker from putting Alice's puzzle pieces together

Once Alice does this, Malfoy cannot impersonate Alice by proving things with her credentials, in either existing or new relationships, because he has no devices that Alice has authorized. This is true no matter which passwords are cracked, which biometrics are circumvented, and which storage collections or secure enclaves are read via forensic tools or side channel attacks. Malfoy could expose every password, credential, and secret that Alice possesses, and still not be able to impersonate her when proof is requested.

Device revocation is not a concept unique to Sovrin. However, an implementation that is secure, global, fast, privacy-preserving, and integrated with digital credentials is a new invention that does not exist in any other SSI ecosystem.[4]

---

[2] Technically, Alice is revoking the privileges of an agent, not a device. A device might run multiple software agents. However, for the current discussion we'll keep things simple and equate agent with device.

[3] Sovrin advocates never allowing a single point of failure in terms of control. Alice can achieve this by having a tablet and/or cloud service in addition to a mobile device. She can also use paper and/or people to exert her control. Such additional points of control should be combined with an M-of-N signature policy to make sure an attacker cannot revoke Alice's devices if he gains control of only one of her devices--whereas Alice can revoke because she still controls a preponderance of her agents. More on this later.

[4] If other ecosystems wanted to adopt such a feature, they would need to support agent authorization registries on their ledger, and they would need to bake into their proof exchange protocol the notion that each proving device must be demonstrably unrevoked.

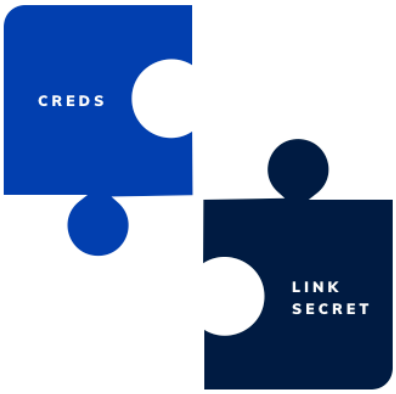
The workings of this mechanism are mathematical and cryptographic, and are documented in a report about decentralized key management systems produced under contract for the US Department of Homeland Security. Sovrin device revocation was demonstrated publicly in early 2018. It's implemented in a feature branch of Sovrin's Indy source code, and is slated for merging over to Indy's stable branch for mainstream adoption after some enhancements are completed.

Importantly, device revocation is reversible. This means that Alice can use it aggressively, without worrying about being too draconian. If the phone turns up at the bottom of her purse instead, the revocation can be undone.[5]

STEP 2: REVOKE RELATIONSHIP KEYS

Alice should next revoke any private keys that the stolen phone knows. She does this on the same device that she used in step 1, and the experience can be as simple as one button press.[6] Where the previous step kept Malfoy from using credentials to *acquire new* trust as Alice, this step prevents Malfoy from encrypting and signing in a way that leverages *existing* trust to impersonate her.

This step may take a bit longer than step 1 (minutes, maybe), but it can be performed in parallel across all of Alice's relationships, and any latency that Alice experiences is likely to be latency that hampers an attacker as well. Also, if Alice and her connections use well designed Sovrin software, the need for speed with these notifications will be low. This is because both parties will have a habit of re-requesting credential-based proof from their remote connection partner--either on a schedule, or at any moment when trust is particularly vital.



Revoking relationship keys subtracts another piece from the puzzle

[5] Reversal is governed either by a key that Alice doesn't keep on her device, or by an M-of-N multiparty digital signature, or both. Thus, reversal is straightforward for Alice to initiate, but cannot be gamed by Malfoy.

[6] Underneath, if Alice is using DIDs rooted on the public ledger, Sovrin software must write new DID docs that deauthorize the suspect keys, as a series of ledger transactions. If she is using peer DIDs, the software must do analogous work to tell each contact known to her phone's wallet that her side of the relationship has been updated.

Do We Need to Do More?

Let's step back for a moment and consider where Alice is protected, having taken these first two steps to remediate.

- Malfoy cannot impersonate her in any existing relationships, or in any new relationships, *ever,* on the basis of anything he gains from the stolen device.
- Alice can continue to use her existing credentials in existing relationships with exactly the same trust as before.
- Alice can create new relationships and build trust in those relationships using her existing credentials, with no risk that the credentials will be misused or that she will be impersonated, anywhere.
- Alice can request and receive new credentials that are just as secure as the old ones. An attacker has not gained any new ability to pose as Alice to request new credentials.
- Alice can continue to rotate keys, authorize and deauthorize devices and agents, and so forth.

If Alice were not using Sovrin, *this list of assurances would be much weaker*: her credentials would be abusable in existing relationships, and could also be abused to impersonate her in new ones. She'd be forced to get all her credentials revoked and reissued, and would be vulnerable for as long as it took for that process to complete.

So, Alice can feel somewhat reassured. Because of Sovrin's unique device revocation feature and the specially designed proving protocol that complements it, t*he immediate risk has been mitigated, and many of the long-term negative impacts vanish.*

But should she take further steps?

ALTERED INCENTIVES

Before answering that question, it is worth considering how radically Sovrin alters hacker incentives, with just the recipe we've described so far.

- **First**, Sovrin eliminates central troves (either in an identity database, or for peer DIDs on a blockchain) and shared private keys. This takes bulk hacking off the table--the most an attacker can hope for is one persona per cracked key, or one person's identity per fully unlocked wallet.
- **Second**, because of its 3-dimensional identity model, Sovrin partitions identity into pairwise relationships, and limits the scope of a hack to only those relationships that a given device knows. If Alice always uses a work-oriented profile on a laptop, and a personal-oriented profile on a phone, then the missing phone only endangers the subset of Alice's identity that's personal.
- **Third**, Sovrin breaks the association

Sovrin protections use all 3 dimensions of identity

between possessing credentials and being able to prove anything with them, such that capturing credentials in and of itself gives no advantage. This drastically alters the need for credential revocation and reissuance.
- **Fourth**, Sovrin shrinks the window for credential abuse *to seconds* from the time a risk is discovered.

This is not a claim that Sovrin's security is impervious.

Although Sovrin has had at least two independent security audits and two years of exposure on HackerOne, no system is hack-proof, and the SSI space is young enough that we should assume vulnerabilities and patches await discovery. However, shifting incentives has substantial value because it keeps malicious attention elsewhere. This is a good thing for Sovrin-centric Alice.

But let's assume that Alice is unimpressed by our side note about shifted incentives. She's wondering what else she should do to be prudent.

WALLETS

When Malfoy steals Alice's phone, he gets a copy of her wallet. What mischief can he do with it?

At first, not much. All Sovrin wallets share a common encryption layer that has very strong guarantees about data at rest. Whether the wallet is persisted to disk using sqlite drivers, or using files in the file system, or using an industrial strength database, the data in the wallet's secure storage[7] is equally mysterious to Malfoy. He cannot use a byte editor or a hacked sqlite engine or admin access to a database or root access in the OS to circumvent this encryption, because *the encryption is independent of the storage layer*. As long as the encryption remains in place, Malfoy doesn't know;

- What is contained in each wallet item
- Which items in the wallet are associated with DIDs, keys, or other information
- How records are indexed (lookup keys are encrypted in a non-correlating way)
- Which DIDs Alice controls
- What public or private keys values might be present
- Which key values are associated with one another or with DIDs or payment addresses

Furthermore, since Sovrin wallets never pass secrets across an API boundary, no software that uses Sovrin wallets could have accidentally leaked a private key to any other place Malfoy can observe.

[7] Wallets are sometimes used as a convenient store for non-secret data as well. This might include app configuration, or a tiny subset of tags that facilitate searches. Such data would be visible--but it would be information that Alice already decided she didn't need to protect. She has to go out of her way to create data like this.

Malfoy might be able to inspect storage and make educated guesses about the data type of certain items, based on size. He might be able to see some dates or numbers from low-value metadata, without being able to ascribe any meaning or context to them. That's about it.

So, can the wallet be cracked?

Of course. The way to do this is not by smashing through its armor, though--it's by going through the front door. All wallets have to be unlocked, and Sovrin's are no different. Thus, Alice's wallet is as safe or as vulnerable as she has made it by protecting her wallet's unlock key.

The strongest protections on the unlock key are provided by a combination of a secure enclave and biometrics enforced by the OS. If Alice uses this type of protection, Malfoy may be out of luck forever. But let's suppose she does something much weaker, and Malfoy is able to unlock her wallet after a minute or two of tinkering.

Now, Malfoy can see every DID, key pair, and credential that Alice's phone knew about. This is a bit of a privacy violation, in that he now knows the eye color on Alice's driver's license and the job title in Alice's employment credential. He might also be able to say with confidence that Alice is connected to at least 203 other DIDs with values (but not owners) he knows, and he would know which public keys have been used to authenticate those DIDs. He might know some of Alice's payment addresses, and which private keys formerly governed Alice's spending.

But as we said earlier, because of Step 1, Malfoy can't use these credentials to impersonate Alice, and because of Step 2, he can't contact any of Alice's relationships and encrypt and sign using the private keys he's found.

Getting into the wallet makes little difference.

LINK SECRET

One other piece of interesting information may be exposed by a cracked wallet--Alice's link secret. This is a large random number that Alice uses to prove credentials belong to her instead of to someone else. Alice also uses it to link multiple credentials in a complex proof: "Here is proof of age from my passport, and proof of residence from my utility bill--and I can demonstrate that the credentials belong to me and can be stitched together because they both contain blinded values that derive from the same link secret that I alone know." It will be in the wallet on Alice's phone if Alice does proving there.

It could be argued that by definition there must be some risk to exposing this value, since it's called a secret. And that is true. But "secret" here has more to do with privacy than cybersecurity; it is a value that is hidden from credential issuers and verifiers to prevent correlation.

Although it may sound dangerous for Malfoy to know this secret, the consequences are not dire. Because Alice revoked her device in step 1, Malfoy has no proving ability just because he knows it. He can't construct new, simulated credentials that appear to be owned by Alice. He can't take the secret out to the dark web and use it to track all the credentials Alice might ever possess in the future, since the secret is always randomly blinded before it's embedded in each credential. He can't correlate it to any of Alice's proving interactions in the past or the future, since it is only a randomly blinded form of the secret that is ever learned during proving.

Alice can continue to use this secret with existing and new credentials, even if Malfoy knows it.

Preventive Measures

Veterans of security analysis may be eager to point out corner cases in which the simple story we've provided above isn't satisfying. And they would be right to do so; corner cases are the acid test for robustness.

Perhaps Alice has no other devices besides her phone. Perhaps Malfoy captures her phone and has hours to hack it before Alice notices the breach. Perhaps Alice has a pending stock trade worth a billion euros, secured until a few minutes ago by a key that Malfoy has now extracted from her wallet. Perhaps a hundred other things.

Sovrin provides additional tools to help. However, many of them should be used before the disaster, not after. For example:

- Some of the keys that Alice authorizes to act in her behalf can be keys she keeps on paper, inscribes on stainless steel, or otherwise persists in a non-digital world. Some keys can be controlled by trusted friends, family, a bank, or a lawyer. Some keys can be built partly from seeds supplied by biometrics. All of these options are particularly important for vulnerable populations such as refugees, who may not have a big digital footprint.
- Alice can set up rich M-of-N authorization policies, such that a quorum of entities she trusts must agree by multi-party digital signature to changes in authorization, to key rotation, to enrolling a new device into her sovereign domain, to payments above particular thresholds, or to other dangerous events. If Malfoy gets her phone before she realizes it, and attempts to transfer a billion euros, he may be defeated because Alice's cloud agent and her laptop do not concur. Malfoy may have a similar failure if he attempts to change Alice's DID document, or revoke her other devices, in a way that wrests control away from her. In an M-of-N scheme, N can include agents belonging to Alice, keys written down on paper and stored in a vault, and keys held by Alice's family, friends, or trusted human agents. Thus, M-of-N can be applied even if Alice only has one device, and that device is lost. It can also help when all of Alice's devices are destroyed in a house fire (because her human friends and paper keys can be offsite).
- Alice can make her authorization policies even more sophisticated by requiring biometrics to complement other key types, with a particular freshness: Only allow a money transfer if one of the keys authorizing the transfer is a biometric that's been provided in the past 5 minutes.
- Alice can define policies that take into account the stakes of an action: any key can authorize a transfer of 20 euros or less; three keys are required for transfers greater than 1000 euros.
- Alice can request in advance from her contacts that they echo back any changes in her DID Document and authorizations as they see them, when they occur. This is a sort of tripwire whereby Alice will learn about an intruder the minute he attempts to alter her relationships in a way she didn't initiate. Designs for Sovrin's relationship management protocol contemplate this reporting as best practice.

- A related tripwire is configurable among Alice's agents. If she has set up wallet synchronization among agents on her phone, laptop, and cloud, her laptop and cloud will know as soon as her phone's wallet changes state.
- Alice can use Sovrin's secure wallet backup feature to keep a copy of her phone's wallet in a safe place, so she knows the exact scope of a breach.
- Alice can pre-arrange for "social recovery"--friends that each keep a shard of her data, that she can contact in event of a disaster. If enough of her friends (M of N) agree to recover her identity, then their pooled shards can be combined into a new wallet that contains keys preauthorized (in Alice's other policies) to take highly privileged action to lock out intruders and finalize a recovery.
- Alice can pre-arrange with her connections to use a "dead drop" as a rendezvous point in the event that a connection goes stale. This mechanism, which is also contemplated in Sovrin's relationship management protocol, prevents Malfoy from locking Alice permanently out of her relationships; if he and Alice are struggling for control, Alice can use the dead drop as a way to reestablish trust with her connections.
- Alice can use Sovrin's relationship protocol to pre-arrange a key rotation policy on a per-relationship basis, announcing that she plans to rotate keys every week, day, or hour. This keeps such relationships on a timeout; if Malfoy cannot wrest control of them from Alice before the deadline, Alice knows they will become inert and harmless. (This is particularly useful in combination with the dead drop for re-establishing the connection.)
- Instead of having a potentially deadly policy like "put everything in my trusted cloud", or an almost equally dangerous one like "replicate everything I need to use my identity onto all my devices," Alice can be very deliberate about which credentials she places on her phone, which agents she permits to hold her link secret, and which agents she uses in which relationships.[8]

It is important to note that preventive measures and recovery features have a potential downside. Each represents an alternate way to exercise control, and thus a new surface for attack. The world's easiest lost password recovery may also be a hacker's favorite place to probe. For this reason, the features listed above strike a balance between safety and ease of use.

Most of these preventive measures are careful applications of the principle of diffuse trust. Sovrin's tools, APIs, trust policies, and best practices carry this principle much further than most SSI solutions, and *light years beyond centralized identity*.

[8] There is work proposed in the Sovrin community to create such safeguards as part of wallet features implemented in Indy SDK.

What's Possible, What's Not

Despite all of the features that Sovrin offers to prevent and remediate disasters, there will always be unhandled contingencies. If Alice falls and develops amnesia just as she's about to revoke her phone, Sovrin's answer may be less than fully satisfying.[9]

The ultimate fallback position of all identity solutions is to abandon everything that comprises your identity (credentials, relationships, devices, software), and start over. It is possible to imagine cases where this will be the answer in Sovrin, too--though Sovrin's layers of prevention and protection make the need for this type of fix orders of magnitude less likely than with other approaches.

Even in this most unpleasant eventuality, Sovrin helps:

- Because Sovrin is meticulous about encouraging diffuse trust, because it buys Alice time to react, and because it supports such rich authorization policy, Sovrin improves the odds that Alice can freeze a stolen identity, even if she can't wrest it back. This is incredibly valuable; it removes impediments to starting over, and makes Malfoy's investment less likely to pay off.
- By focusing on zero-knowledge proving and attribute-based trust, Sovrin makes a restart of identity feel very different for connections. They may need to substitute a new DID for an old one, but they don't need to map from old credentials to new ones, because they don't ever see your credentials in the first place. They simply perceive that you are using a new DID with your familiar attributes. Sovrin's credential revocation is testable against specific points in time, such that you can use a revoked driver's license to prove you were a licensed driver before the revocation event. This may eliminate or at least postpone the need to restart in some cases.
- Sovrin's credential revocation (a way to invalidate a credential because the issuer no longer stands behind it) is testable against specific points in time, such that you can use a revoked driver's license to prove you were a licensed driver before the revocation event. This may eliminate or at least postpone the need to restart in some cases.
- Sovrin's credential revocation mechanism is uniquely privacy-preserving--so a restart does not leave a trail of breadcrumbs for third parties to analyze, compromising privacy on either your old identity or your new one. This contrasts with other approaches, where an attacker may be able to perform temporal correlation on revocation lists or similar resources, learning more about your identity even as you revoke it.
- Sovrin lets you start over on a per-relationship basis rather than across all relationships, if you like. It lets you start over on a per-credential basis rather than globally, if you like.
- It lets you start over on a per-device (per-agent) basis rather than across all devices, if you like.

---

[9] Sovrin *still* helps. Its governance, ledger, and software include the scaffolding for guardianship, which could rescue unconscious Alice. If you are interested, contact Sovrin about attending its Guardianship Working Group.

- It requires all Sovrin wallets to support an export and import feature, guaranteeing that your credentials and relationships are never hostage to a particular service provider. This also guarantees that Alice can know exactly what needs rebuilding.
- It provides free, open-source reference implementations of the full software stack, guaranteeing that your software is never hostage to a particular vendor.

Conclusion

Sovrin has a very rich story for disaster recovery. Common problems that would be disastrous in other solutions are quickly remedied and/or substantially mitigated in Sovrin. Even for more complex challenges, Sovrin provides excellent preventive measures and a set of strong guarantees about how recovery can be managed. These measures are a reflection of the principle of diffuse trust, which is a core tenet of Sovrin's architecture. Sovrin cannot eliminate all disasters, and its solutions are still maturing. However, it is clearly a thought leader on this topic in the SSI community.