2021-01-20

Where:https://zoom.us/j/4034983298?pwd=STZQd0xMZU9xRVVOVnpQM3JNQ2dqZz09

When: Jan 20th, 12:00 pm EST (17:00 UTC)

Details: https://wiki.hyperledger.org/display/IWG/2021-01-20 - This page

Attendees:(15)

Vipin Bharathan

Dr. Samuel Smith (https://keri.one)

Jim Mason

Luca Boldrin

Stephane Mouy

Daniel Bachenheimer

Bernt

Audrius Ramoska

Charles Lanahan

Christos Patsonakis

David

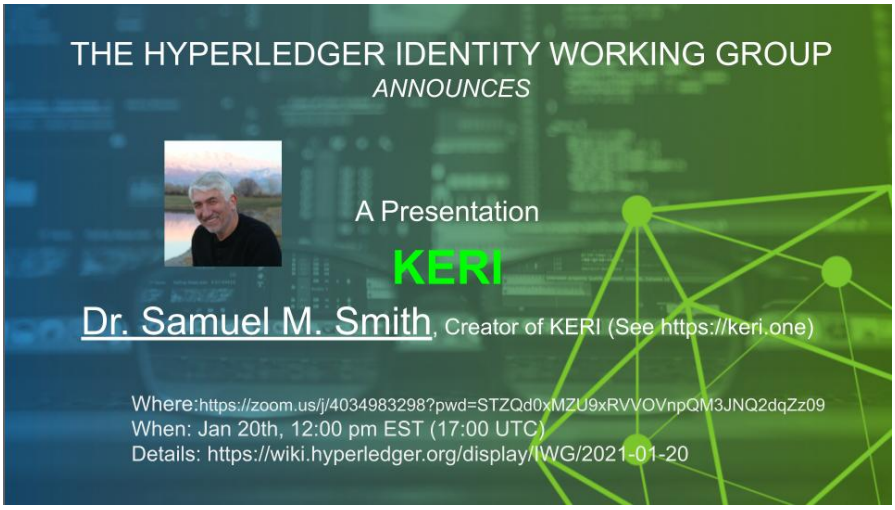
John Wick

Mark Scott

Main Event:

A presentation on **KERI** (Key Event Receipt Infrastructure)

Dr. Samuel M. Smith, Creator of KERI (See https://keri.one)



KERI White Paper

The long form (140 pages)

Recording:

VIDEO

Recording of an earlier talk at the Human Colossus Foundation

Response to follow up questions

Questions

We delved into one of the techniques for preventing duplicity, namely "an eye of God" or as we concluded the distributed eye of God.
We did not go into detail on what such duplicity can achieve, please clarify what is in it for anyone to be duplicitous? Unless of course there is a stolen private key which can then create these alternate forward keys in the KEL.

Also, you mention the following two points in your contribution to Chapter 10 of the Manning SSI book.
How does KERI solve the problems that you detail below?

1. Stolen private keys
2. Not noticing the theft before assets are drained or some other form of harm happens.

Digital keys can be stolen remotely. Stealing a physical key requires having physical access to where the key is stored or who is carrying it. Digital keys that are not properly protected can be stolen remotely over a network. Even when they are
well-guarded, digital keys can still be stolen using side-channel attacks2 (but those are very hard to pull off).
You may not be able to tell if a digital key has been stolen. A stolen physical key is easy to spot (unless the thief can quickly copy and replace it—a real challenge). But if an attacker is able to gain access to a digital key, they can copy it in milliseconds without you ever even knowing.

Answers

KERI provides a primary protection from side channel attacks via pre-rotation. A pre-rotated key is essentially a cryptographic commitment to a key that is not and has not ever been used to sign anything. Therefore it is not exposed to a side-channel attack on event signing infrastructure. So stealing a pre-rotated key must be an attack on the key creation and storage infrastructure not the event signing infrastructure. The key creation and storage may be air-gapped so that it is not exposed to internet attack. The pre-rotated keys are not used to sign anything so being air gapped is not an impediment. The first use of a pre-rotated key is to sign a rotation event. The rotation event itself can be signed while air gapped and then moved to a computer that is connected to the internet to publish the event. At which time it is now two late for an attacker because the first seen (published) rotation event is the authoritative one. Any observer first sees that authoritative one and a latter alternate made by compromising the now exposed key pair is too late.

The eye of God is a **community of observers** that share what version of an event is their first seen version. If publication of a rotation event is shared amongst a suitably large number of observers who are honest then no later attacker may change the consensus view of that community. Honest observers follow the first seen rule. The first seen verfied version of any event may not be changed its first seen status is immutable. Any other versions seen later are marked as duplicitous.

So a successful attack on such an observer network requires several difficult feats by the attacker. It must compromise a pre-rotated key. This is a key that has not been exposed. This may require a physical attack on air-gapped storage. If that pre-roatated key is a multi-sig set of keys each stored in a different location, then it requires multiple coordinated simultaneous physical attacks.

Attacks on the public pre-rotated key become easier once it is exposed but now the time window for success is very small. Once published the original version of the rotation will be disseminated across the observer network in milliseconds or at worst seconds. So a compromise of the private key via any other attack after exposed must occur before widespread dissemination or else its too late. One way to extend that time window is to mount an eclipse attack on the observer network. But this is very difficult to do in general and if even one copy leaks out before there is time to compromise the private keys then the attack fails in general. It still might work on some small subset but then only for a short period of time as eclipse attacks are detectable. All observers have to do is send signed heartbeats to each other (i.e share signed messages of any kind) and not get back acknowledgement or receipts of those messages to know they are being eclipsed. which means they should escrow all rotations until such time as the eclipse ends. So trust in any signature requires making an appraisal on the state of the observer network as a function of the network propagation time. If a statement that is part of some transaction is signed with a key from a rotation event and that statement has been committed too in the authoritative key event log by an event after that rotation event then the statement is valid. The authoritative key event log is the first seen version by some majority of trusted observers. Each party gets to pick which observers to trust. If there is duplicity and the party is not able to resolve that duplicity into a single authoritative log. Then it trusts no signatures until such time if ever that the duplicity is resolved.

KERI is not meant to provide double spend proofing. Its means to provide secure attribution. KERI guarantees look like, either I can make a reasonable choice to trust or i must not trust at all. Any duplicity means do not trust until you can resolve the duplicity. So you don't engage further in a transaction. The transaction stops. If you detect the duplicity after the transaction has completed then its too late. So highly valuable transactions should have multiple steps with time delays long enough for all events to propagate network wide so that duplicity is resolvable. Because each message is signed with a non-repudiable signature, any fraud is provable. All that can be said is that one's keys have been stolen but that does not absolve the party from liability. they are liable for managing their keys and are responsible when they are stolen. If a party digitally signs something and there is an authentication process at presentation of the party who purports to be the controller of that signature then that party is now liable for harm to the other party arising from that. If the other party does not authenticate the controller of that signature (like with a notary) then liability is a function of the rules of the road for signatures on that type of transaction. But notarizing a signature is usually enough to put the liability on the signer and notary for a fraudulent digital signature not the other party. This is why digital signature notaries carry liability insurance.

There is no free lunch here. KERI just makes it so roots-0f-trust for digital signatures and the associated identifier systems may be fully decentralized and duplicity detection protects validators. Validators only trust the controller of a root-of-trust if and only if they can resolve any detected duplicity enough to trust by their standards of trust. Not the controller's.

A controller can use the observer network to see if its keys have been stolen by looking for duplicity. Resolvable duplicity or recoverable duplicity means the controller performs a rotation to recover. If the duplicity may not be recovered from then the controller must abandon the identifier.

The only thing that is not recoverable is a compromise of the pre-rotated keys (at leas for non-delegated identifiers). If the pre-rotated key is compromised before the original controller first uses it to publish a rotation and the attacker creates an alternate rotation and publishes it first then the controller must abandon the identifier. But the attacker must publish first making its attack immediately detectable. So the controller an minimize harm because it immediately abandons that identifier and no longer uses it.

KERI never fails to detect a failure and make many failures (except one) recoverable. This is core to good security. As you well know the typical exploits go undiscovered for months. In KERI an attacker must not merely compromise keys but also be the first to publish use of those keys. If they are second then they are too late as the first version is the authoritative version not the attackers later version. But as soon as the attacker publishes then the original controller knows of the attack and may then immediately abandon the identifier. So any multi-step transactions becomes very very difficult to attack because success requires all steps be attacked in sequence without detection but a given step may not be accepted until its published and therefore detectable so the attack is thwarted after the first detected step.