TRANSCRIPTS

# May 2026 Videos

A complete transcript of every recorded talk from the May 2026 Videos.

| 1 | 48:48 |
| --- | --- |
| TALKS | RUNTIME |

November 26, 2026

# CONTENTS

Full transcripts of all 1 recorded sessions, in the order they appear in the archive.

01 2026-05-05-Introduction-to-KERI-Part-1-XYohZTh3yEw 48:48

TALK I

# 2026-05-05-Introduction-to-KERI-Part-1-XYohZTh3yEw

May 2026 Videos · talks · 48:48

Welcome. My name is Samuel Smith. I'm the creator of the KERI suite of protocols. And this video is a tutorial talking about KERI from a high level. KERI stands for Key Event Receipt Infrastructure. It's been in development for several years now.

And you can think of it and as a secure identifier overlay as a trust spanning layer for the internet. So, we're going to talk about what that means today. For resources, you can go to the website KERI .one. Should be easy to remember. Look at the resources page. There is an active KERI community with various standards and specifications. You can find the code, which is is Apache 2 licensed at the Web of Trust GitHub repo. The Trust Over IP Foundation, which is a member of the Linux Foundation's decentralized trust organization, governs the specifications for the relevant protocols. These are licensed using the Open Web Foundation license, which is a freely licensed open specification license. And you can find meetings there and there are a couple of links for that. The motivation for KERI came from years of working with protocols. I started life as an academic, did a lot of work with AI and autonomous vehicle systems back in the 90s. Started working with protocols in the Internet of Things also back in the 90s. And later in my career, started working on cloud-based automation systems and was confronted with the problem that the internet did not have a really good security system. And if you think about it, the internet has lots of problems today and they're getting worse.

And so this idea that the internet was broken in can be traced back to the fact that the internet never had a security layer. If you look at the comparison of the OSI model and the IP model, the layers where you do authentication are in the presentation and session layers and the IP protocol doesn't have those layers. The result is that instead we use bolt-on security systems or sometimes called security overlays These are commonly known as domain name systems, certificate authorities, openID connect, OAuth, various types of systems that provide security. And that security is based on what was a good idea 20 or 30 years ago when it was created, but now is no longer such a good idea. And so KERI is about reimagining what the internet would look like if we had a security layer that actually worked.

The idea can be traced back to something called a spanning layer. So, the internet in its early days wasn't viewed by everyone as something that was going to be universal. It was an experimental protocol. But very quickly, relatively speaking, it dominated all of the other communication protocols and became the dominant protocol for the whole world. And one of the analyses as to why that happened is that the internet had a spanning layer and that spanning layer made it possible for the internet protocol to grow in a way that allowed it to subsume all the other protocols and provide all the services. And that what that spanning layer in the internet is the IPv4 for, you know, IP version 4 and now we have IPv6. But originally this is the diagram that was used to demonstrate that there was something called a spanning layer. And if you think about it, it looks like an hourglass and the waist of the hourglass is the IP layer. This is the addressing layer. Everything above that layer uses the same address and everything below that provides support for that addressing layer and that feature of the spanning layer. And you can look at some of the papers that have been written about that is reputed to be the reason one of the reasons the internet became so successful as a protocol. The problem that we've had is that because the spanning layer of the internet did not have any security built into it, it didn't have a trust basis. What has happened is that we now have what I call platform lock trust.

In order to have trust, you have this IP spanning layer and then you have supports for it. And then you have applications built on top of the spanning layer, but they don't have trust in them. And then you bolt on trust above those. And so what you get is trust domain based segmentation of the upper layers. And so you don't have trust across those those walled gardens or columns above that. And so you don't have trust that spans the internet. It just spans a given platform. So you get a bifurcated internet trust

map. There's no trust spanning layer in the current internet. And you might think of this as like, you know, different platforms that have different trust bases or, you know, and often trust is the identity system, right? So you have a Facebook identity system and within Facebook you have trust and on Google you have its identity systems and within Google you have trust and you might have other platforms like blockchains that have trust within their trust domain but they're not interoperable. You can't move trust between those trust domains. So the basis of the spanning layer, there's a really well-written paper that talks about the hourglass approach to building protocols where you have a spanning layer. And so we feel that that's the right approach, but we just need to have a trust spanning layer. And we don't want to reinvent a new internet that would be ludicrous, but we want to allow the internet to remain the internet, but add a spanning layer that allows trust. So there's not locked to different platforms. And so our solution we call a waste and a neck is we have the IP spanning layer, which is the waste, and then we add a trust spanning layer that is the neck. And if all trust spanned applications use the same trust spanning layer, then they'll have interoperability in terms of their trust. And so you can still have platforms, you can still have all the protocols down below, but if you want to have applications that span trust, where trust spans across the applications, then we want to have a trust spanning layer. So that's the design goal of KERI.

So I've used the term trust basis or trust domain a lot. And so I just want to define it so that you understand what I'm talking about. A trust basis is the set of trust roots. And when we talk about systems design, we talk about things like roots of trust, things that you can rely on that have certain well-defined properties. And then based on those things, you can build other services and functions on top of that. So we talk about a trust basis and then a trust domain is the ecosystems of interactions that rely on a given trust basis. So for a trust spanning layer, we would need to have everything above that rely on the trust basis that is the spanning layer. And so KERI is about defining what that looks like. So, the basic idea of KERI, and we'll go back into this in more detail, but the idea is that we can define an identifier system security overlay. It's a new type of overlay that acts as the trust spanning layer. So, you have infrastructure that sits below the secure identifier overlay or the secure identity overlay. And then you have a trust domain that sits above that. And the spanning layer is that overlay. Now, when people look at KERI, often they look at it and it's difficult to understand. And one of the reasons that KERI is difficult to understand is that we call it the platypus of the identity space. And the reason for that is if you look at the platypus, it's a very interesting animal. It's difficult to classify because it has a unique combination of features not found elsewhere in nature. or at least not in that combination. For example, because it has milk, a platypus looks like a mammal, but it also lays eggs, so it looks like a reptile. It has a bill, so it looks like a bird. It has electrolocation, so it looks like a ray. And it has venom, so it looks like a snake. And so if you approach a platypus from the mindset that every identity system has to look like a mammal, it will be confusing. Or if you approach it from any of the other viewpoints. So KERI is best approached from the fact that it is truly unique in many important ways.

So KERI deconstructs what many people are familiar with in identity and security systems to create a novel combination of properties that solve what we call the universal secure attribution problem. And we'll go into that in more detail. If you think about a system design trade space, you can have lots of puzzle pieces that you can put together and there's lots that one can select from. But KERI carefully selected a subset and left certain things out that it didn't need so that it got the properties that it did need to solve the problem and the problem of a trust spanning layer requires solving the problem of universal secure attribution.

And if you think about it, if I want a trust spanning layer, then I have to be able to do something universally. So KERI is a well-fit-together puzzle where the pieces are carefully designed to match that trade space. So let's step back a little bit because we've talked about the internet. Let's talk about each individual person. Why should I care about having a trust spanning layer on the internet. Why should it matter to me? The fundamental idea that KERI tries to solve is to enable entities, both natural persons and legal entities of all types, to retake control of our data and the value of that data. And if you think about the internet and how it's evolved, most of the value of the data on the internet is not controlled by the individuals that source the data or or subjects of the data, but they're controlled by large internet companies, super aggregators, super data correlators.

They control the value of the internet. People gain a lot of value of interactions, but many of those interactions are designed to maximize engagement and often engagement is maximized through rage or other deplorable human type interactions. And so we want to have high value interactions on the internet that are valuable to the people interacting. They're not being exploited.

So when you think about building a system, you think about, well, how do I build something? What are my tools? What are the toolkits available to me? And so if my toolkit, and these are tools from my tool chest in the garage, these are all hammers, various types of hammers. and they work really well with nails. If I have a set of screws, those hammers aren't very useful. I need a different toolkit. I need screwdrivers. So we need to figure out what our toolkit is because actually in security systems, we don't have a choice of toolkits. We've got one. So what we have to do is we have to figure out how to make it work with that one toolkit and we have to be able to build our systems with just one So if our toolkit is screwdrivers, we've got to build everything with screws. If our toolkit is hammers, we have to build everything with nails. So there's really only one set of tools for truly secure data control. We only have one. And so we have to figure out how to best use that one. And I would say that the existing internet security systems don't do a really good job of the existing toolkit. Now, fair enough to say that 20 years ago, ago, the toolkit was even worse than it is now. They had fewer tools and they did the best they could with the set of tools they had then. But we have better tools now, but it's still of the same type of tools and we could do better than they did. And so KERI is about recognizing that the tools have gotten enough better that we can do significantly better in our toolkit. And those tools are entropy-derived tools. So these are primarily cryptographic one-way functions.

So these are things like cryptographic strength pseudorandom number generators, hashes or digests, the ECC key pair generation algorithms, digital signatures, encryption. Now, in a few years, we're going to have to add another set of tools because quantum computers are going to provide the ability to tax ECC key pair generation. And so we need to have post -quantum key pair generation algorithms, which NIST has now approved. So it has to be on a roadmap. But for today, we can just focus and KERI was designed five years ago. So we can focus on that, but KERI is cryptographically agile. So KERI doesn't have already has a migration path to be post-quantum secure. So let's talk a little bit about the economics of data. As I said, we want to we want to be able to retake control of our data and its value. So when you think about economics, it's about value, right? And it's about the creation and capture of value. And creation and capture looks like extraction, exchange, and exploitation, both in bad and good ways. And so if we want to take control over the value of our data, data, then we need some way to secure that control. So the goal is control over the value. And there are two types of value. There are its, these are atoms. People are built out of atoms. So you can think of people as value of people as also atoms. We have things built out of atoms, buildings, cars, things. And to secure those atoms, we have

something called physical security. And there are lots of physical security systems and tools for physical security. And for the most part, it's pretty difficult to attack physical security. It usually involves putting lives at risk to ultimately gain physical control over atoms that are protected.

Physical security is largely a solved problem. The other type of value are bits. This is information value. And the problem with information is that it's digital and can be copied. So you have documents, you have places to store data, but you can copy information at will indefinitely without losing any of its fidelity.

Whereas we don't yet know how to copy atoms. We don't yet know how to clone people. It requires new atoms. You can build things that look like old things, but they're different things. And the cost of building them is high. Whereas information, to make a digital copy of information, the cost is extremely low.

And so information security is by nature a different type of security because we can't have a single copy that we protect physically. We have multiple copies. And once those copies leak out, then they're everywhere. So, and then the question is, you know, what does that look like? You know, from the standpoint of people, you know, how do people control, you know, how do we have information security? So, we want control over value. We have the two types of security. And the one thing that we have, in our toolkit, and remember I said they're entropy-derived tools, is that we can have digital uniqueness. It's a very narrow type of digital uniqueness and it's called entropy. And basically we can create numbers that are derived randomly such that somebody can't guess the number as long as the number is kept secret. So if we don't share the number, they can't guess it. And that is the only security that we have. Everything else is one-way functions that try to preserve that original entropy, that original level of cryptographic security that says they can't guess the number. And so that's not a very rich tool set. So now how do we control things? Well, we as people, we are its. We want to control bits. How do we do that? Well, the way we do that is that we create cryptographic pseudonymous identifiers that are derived from entropy and we build a mechanism that allows us to prove that we control those identifiers. We're not talking about all the information in the world. We're talking about just identifiers. We can control identifiers. And if we can control identifiers, And then if we have a way to make attribution to information to an identifier, then we can at least assert that the source of the information came from the person or entity that controlled the identifier. That still doesn't sound very satisfying, but that is all we have.

So, we have to build from that. So, let's think about it again. to retake control of our data, we will first retake control of our identifiers. Now, this is where the internet falls down. If you have a domain name, you're renting it. You don't own it.

You're renting it from a service provider. There's a limited number of those domain names, so they're a scarce resource. And so nobody can own it. They just rent it. All right. Now, you can maintain rental control indefinitely, but it's still not yours. If you create an account on some social networking site like Facebook and you get an account there, you don't own that account. Facebook owns that account.

You can create a username and a password, but you don't own what's on that account. And that username and password, they can at any point in time say, we don't want you on our account and they can remove you from the account. And so you don't have control over the data and you don't have control over the identifier. So since the only thing that we can have information security over are group graphically derived identifiers, we must first retake control of our identifiers. So that's what KERI does. KERI says, we're going to develop a system where each individual controls their identifiers and hence

their identity. When we talk about identity, we're going to narrowly think of it as identifier plus attributes and anything that can be securely attributed to that identifier. That's the identity. So it's a very narrow definition, but it's the only definition that actually works. If we define it more broadly, then we don't have control over it. And so there's lots of differences in identity. So I tend to avoid identity in general. And when I say identity, I really mean an identifier system security overlay, or I might call it an identity system security overlay, but it's really about the identifiers. So identifier uniqueness comes from captured entropy. Anybody can capture entropy of sufficient strength that there is no ability to attack it. Even a quantum computer can't take that away from you if you do it right. So what we create are cryptographic pseudonymous identifiers. Those are the types of identifiers. We call those cryptonyms. That's short for cryptographic pseudonym. And the hard problem is how do we provide provable perpetual control over that cryptonym?

That's the hard problem KERI solves. It's not ephemeral, not temporary, but perpetual control over a cryptonet. If you don't control your identifier in perpetuity, then it's not a very useful identifier. Now, I want to go from the identifier to the data. If we have the ability to have provable control over identifier, what we would like to do is we like to have control over data associated with that identifier. So the first thing we want, and I call this the four A's of secure data control or A raise to the fourth power data, right? So we need an author of the data, the creator, the source of truth, the root of trust for that data. And that imbues the author is the only one with the authority to control that data, right? So authorship and authority are if you can establish that, then you've created the basis of secure secure control. Now you need to be able to authenticate any action that you perform.

So if you create data, you need to be able to prove to somebody else that the origin of that data was from you, the author, right? So you need to be able to do secure attribution. You need to be able to provenance the data. When we talk about provenancing a painting, right? What's the provenance of a masterpiece? Well, the way you provenance it is that you trace the ownership of that painting back to the original painter, the original author of the painting, the painter. If you don't have a chain of custody of that painting all the way to the present day, then you can't securely establish that the painting is not a forgery. It may be a forgery.

Forgers are really good. You can look at the painting and you go, well, it doesn't appear to be a forgery. There's no obvious things that make it a forgery. So that's a good sign that it might be authentic. But without the provenance, the chain of custody all the way back to the painter, you don't know for sure that it's an authentic painting. So we want provable origin. We want provenance. We want secure attribution. And then if we've created data that we can prove is sourced from us, then we want to be able to control what happens to that data. And that means authorized the use of that data, give permissions, consent. It establishes what people call a loci of control for the data. That is the domain over which you have control. So if the data is in your loci of control and you maintain that loci of control, then you have control over the data. And so you need mechanisms to authorize the use of that data or what happens with the data or anything else.

And then finally, a property that is not cryptographic at all is authoritative. For data to have value, it must have a creator, it must have a provable origin, the use of it must be authorized, but the data also has to be authoritative. It has to be credible, accurate, reputable. And so we need mechanisms for that which we can bootstrap on top of the other three.

Let's now go back to the trust basis of humanity when we interact as humans and think about how we interact as humans because we're built really well for those sorts of interactions. Our brains and our senses and how we function, our culture is all has been designed to enable us to function really well

with the human trust basis. And in the human trust basis, we interact with people in person. We see them, we touch them, we hear them with our physical senses. And over time, we get to know people from those in-person interactions. And so that knowledge, right, that we as a human can know other people means that we can then trust them.

And how we trust them is based on those physical interactions, right? As humans, we're interacting. And so we build the ability to trust other people and to trust what they say and what they do, right? And it doesn't mean that we aren't fooled or tricked or people don't lie. All of those things happen. But we have a mechanism by which we operate fairly well that works pretty well, has worked pretty well throughout history. And we have lots of institutions that bulwark that human trust basis. The problem is, is on the internet, we can't really know somebody human use in the human trust basis because we don't interact with them physically. We don't see them with our senses. We don't touch them. So we can't really trust what comes over the internet because for all we know, the bits that came over the internet didn't come from a real person or didn't come from the person that said they came with. They could have come from somebody else or they could have come from an AI or for a bot or from a dog for all that matters, right? It is a much more difficult problem and we don't have the tooling as humans to do that. We have to use different tools. So we have to be able to switch from a human trust basis to the only tool that we have for information security, which is a cryptographic trust basis. For those of you who don't like cryptography, this is probably not the thing you really wanted to hear. But a cryptographic trust base is the only one that we have. And to the extent that we can use a cryptographic trust basis, then we can have really strong information security. To the extent that we use other trust bases, then we're not going to have strong information security. So one of the goals of KERI is to provide strong information security without requiring everybody to know anything about cryptography. You just need to know the best practice for using it and hopefully all the tooling under the hood hides all the cryptography from the person. You know, this tutorial is to lead up to for people to get to understand KERI at increasing levels of depth. So this is really high level theoretical abstract, but we're going to start going deeper as we get further along. So one of the problems we have to recognize is that cryptography has limitations. For example, one of the cryptographic tools that we use heavily is a verifiable digital signature. And it depends on something called a cryptographic public-private key pair. It's also called asymmetric crypto. So with a digital signature, you can take a statement. This is a digital statement.

You sign it with a private key and you can verify it with a public key. This makes it very useful for communicating over the internet. And it enables the strongest form of security because you're not sharing a secret. And what does that mean?

mean? Well, with a digital signature, any signed statement is non-reputable by the controller who signed it. What we mean by controller is the individual entity or natural person that controls the private key, right? Because in order to sign it, you have to have the private key. So, what does non-reputable mean? It means the signer cannot plausibly deny that they created the verifiable signature.

Therefore, we may only verify the controller of the private key and we call this the who that made the statement. So the author, right? But not the veracity of the statement. We can't use cryptography to prove the veracity or accuracy or truth of a statement. And that's the what, right? But we can prove the who to at least the level of plausible deniability. Okay. In order for that to occur, the assumption is that the private keys behind a digital signature, the private key or keys are truly private, never shared, right? And that the signing, the act of signing reflects the signer's true intent. And we call this these two things together, the signing infrastructure. And we say the signing infrastructure was not compromised. So, for

example, if somebody steals your private key, then they can sign stuff and impersonate you. They can say they are who you are and somebody trying verifying it will say, well, it verifies, therefore it must have been signed by the private key. And so if the private key has been stolen, then you lose non-repudiation because now anybody, whoever, multiple people now have that private key. Correspondingly, if I have a computer or a device that uses the private key to create the digital signature, I have to have some way of inputting to that device something to be signed. And so that digital artifact that I input to be signed, if somebody is able to input something to be signed and I permit that signature to happen, maybe it's on my device and they get it onto my device and they tricked me into signing it, then now I've signed something that I didn't intend to sign. And so that's another type of compromise. So the assumption is that either the keys or the signing or the act of signing has not been compromised. If that is true, then digital signatures provide non-repudiation to the signer. And that gives you a cryptographic basis for securely attributing a statement that has an attached signature to the controller of the private key as a source of that statement. So with cryptography alone, we cannot trust what was said. It doesn't provide that. We have to use it to other mechanisms in concert with cryptography, but we can say something about who said it. We can only detect if the signer is being inconsistent or duplicitous. So what that says is if you don't know what was the statement is true or false, but you do know that the signer made the statement.

They're basically saying, I say this. So someone could say the sky is green or they could say the sky is blue. Well, if they're consistent and they always say the sky is green, they may be wrong, they may be misinformed, they may be delusional, but they're not being inconsistent, they're not being duplicitous. If on the other hand, they say the sky is blue one day and the sky is green another day, they're not consistent. Or if they say the sky is green to a set of people and the sky is blue to a German set of people, they're being duplicitous. And if they sign both those statements and you produce a statement that says the sky is green and the statement that says the sky is blue and you see both statements, you verify both of them, now you have provable duplicity or provable inconsistency depending on how you think about it. And therefore, you have reason not to trust the signer. But if you never see inconsistency or duplicity in all of the signer sign statements, then you may be able to trust that signer. At least you know that they're not being inconsistent. They're not being duplicitous. They still may be delusional or they still may be malicious and be a liar, but they're a really good liar because they always use the same lie. And that's something. That's not nothing. That's really, really valuable on the internet because then we can use other mechanisms to hold them accountable for the veracity of their statements. And those other mechanisms now are mechanisms that are already established. We have all sorts of legal, regulatory, and economic mechanisms and incentives that we can apply to incentivize people to have veracity in what they say. And those incentives are enforceable if we can prove who said it and that a given person or entity said something. And so we start there and then we can finish with stuff that isn't cryptographic, but we can't get there over the internet without the cryptography because over the internet, it's just data.

It's not its. So what makes KERI possible? KERI was not possible 20 years ago. It was possible probably only 10 years ago. And actually the development of KERI, I started working in this space heavily in 2015. It was possible a few years before that, but not likely. And KERI was a result of five years of trying to solve the secure attribution problem and failing. and then learning from all my mistakes. And then in 2019, I wrote the first KERI white paper and 2020, the second edition of the KERI white paper, which is largely the KERI we see today.

And since then, we've been building on that. So what happened in, you know, between 20, before 2015 and after 2015? Well, then before 2015, it was generally considered by everyone that users could not manage private keys. What I mean by cryptographic keys, to manage them, to store them, create them, not lose them, use them to sign things. That was too hard. In fact, the DNS certificate authority system and the federated identity, OIDC, OAuth, all of those systems are based on the idea that the average user can't manage private keys. So we get federated identity largely with things called IDPs or identity providers.

Identity providers are enterprises or institutions or entities that have the capability, they have the education, they have the technical know-how to manage private keys. And everybody else is a relying party because they're not managing private keys. So we built a system based on this assumption, the impossibility of user-managed private keys. It doesn't work very well anymore. Now, post-2015, we have mobile devices. We have mobile phones that are essentially supercomputers in your pocket that have devices and, you know, hardware security modules, multi-factor authentication, secure boots, secure enclaves, password management apps. And a lot of that was actually a result of all of the excitement over blockchain that started in the big inflection point, I would say, is 2015. That all of a sudden now, because all of blockchains and cryptocurrencies are based on users managing private keys. And so users were forced to figure out how to manage them and people were forced to develop tooling to manage private keys. And so that generated a whole bunch of effort that resulted in private key management. So when it came to solving the identity problem and creating a trust spending layer, I could say, well, I can assume that users can manage private keys. So I can assume that people can control identifiers that are cryptographically derived from private keys and therefore they can use those private keys to do secure provenance of those identifiers. So now I have the ability to do that. So 2015 practical user managed private keys. And since 2015 15, it's gotten even more practical. I mean, the user experiences has gotten much, much better. People now regularly use passkeys, which under the hood are just public-private key pairs.

This is something that we shouldn't say anymore, that we can't build identity systems where the basis of the identity system is on an outdated assumption, the assumption that users can't manage private keys. That's outdated. We can build them in And we have devices that do that. So then we had to settle for good enough security from federated identity. And federated identity has lots of security problems. Primarily because it's based on shared secrets. And shared secrets are reputable. And now those shared secrets are phishable or spirit phishable or any number of attacks. And so, you know, a shared secret is a password. A shared secret is a OAuth bearer token. Shared secret is a TLS exchanged cryptographic key that is used to do a TLS handshake to provide what is called authcrypt. It's a weak authentication mechanism that's reputable and we shouldn't use anymore.

But the whole internet is based on it. So that meant that with good enough security, you have to have secure sessions. You ended up with perimeter security with VPNs that also use shared secrets because you Every user of the VPN is using the same private key because the IT admin is sending out the same client certificate to all of the users. Now we can have identity based on user managed private keys. And that means we can have a difference in security that is different in class because we don't have to ever use shared secrets ever again. And KERI is that. KERI is a protocol designed to maximize that. So let's talk about private key management. What that means is that you have controlling keys. You have key pairs for a digital signature. The controlling keys are the private keys. You create an identifier. We haven't said how we create the identifier. We'll get into that. But abstractly, you create an identifier from those controlling keys. You now have data and you use that identifier and some cryptography, namely digital

signatures, to establish secure authorship or attribution over that data. The problem is that the continued use of private keys expose them to what are called side channel attacks. What that means is that they're not actually attacking side channel means they're not directly attacking the digital signature. The digital signing algorithm, if you're using elliptic cryptography, especially the most secure ones like ED25519, It has been secure for the last 20 years. And multiple Moore's laws increases in computing power have not materially weakened the security of those digital signatures. So elliptic curve digital signatures are secure. Now, when quantum computers come around, that won't be true anymore, but we already have new cryptographic algorithms that replace ECC that give us the same thing as just the key sizes are bigger and take more computation. But it's still not vulnerable to direct attack. A side channel attack means you're not breaking the crypto, you're breaking the key management infrastructure and you're breaking the signing infrastructure. And so the more you use private keys, the more you use your signing infrastructure, the more likely an attacker is to figure out how to break your use of it, not the cryptographic algorithm, but the way you use it and the systems you use it on. And so what has happened is what happens then is that practically speaking, the keys become weak from exposure. The more you expose them, the higher the likelihood that one of these side channel attacks will be successful. So the way to manage that, so this is the hard problem with key management, is that your private keys need to be revoked and replaced with new private keys. The short term we use that for that is to rotate your keys. You rotate from an old set of keys to a new set of keys. And the new set of keys give you a fresh start that haven't been exposed to CIDESHILL attack. And so now the attacker has to start over, right? He has to do whatever to try to compromise your system. And so if you rotate them often enough, you can stay ahead of the attacker in general. Not always, but that's the general practical thing. The problem is most systems that are based on public key, if PKI stands for public key infrastructure for the PKI system is based on public keys, you must reestablish the root of trust with each rotation because the root of trust is based on the public key, not an identifier that persists, but the public key. So you throw away the public key when you rotate it. Now you have to create a new public key and you have to redo all of the work you have to do to reestablish the root of trust of that public key as being authoritative for a given person or entity. That means the identifier is how you link that person to whatever it is that you want to security attribute. And if you throw the identifier away, which is the public key, then you have to reestablish all of the reputation. All of that has to be reestablished. When you have a compromise, you break the chain of trust of control over the identifier.

So if the private key is compromised, then the public key, the provenance is gone, everything's gone. So you go, well, wow, that's really hard, right? I have to do private key management if I want, if I, you know, to be able to do cryptography.

But now the hard problem is not the cryptography. The hard problem is the management of your private keys, right? So key compromised recovery is how you fix this problem. You need a mechanism that despite your key management breaking down, can you regain control? Can you recover from that? So the hard problem of private key management is key rotation with key compromise recovery, right? So that even if your private keys are compromised or your signing infrastructure is compromised, that you can recover control over an identifier that persists despite key rotation. KERI solves this hard problem. Until KERI came along, nobody solved this problem. The other people have now copied KERI and some parts of KERI.

But in terms of a full system that solves this hard problem, as far as I know, KERI is still the only protocol on the planet that is public and open source and open standard. There may be proprietary ones that I don't know about, but of all the public ones, it's the only one that I know about. That's the end of this

discussion.

So hopefully I've whet your appetite for understanding what KERI does because I basically told you how KERI solves the hard problem with key management. And if we don't solve the hard problem with key management, then we can't control our identifiers on the internet, which means we can't control our data and we can't control value of our data. So if we really want to control the value of our data on the internet, we need to control our identifiers and we need to do key management to do that and to do that we need KERI. So I hope that's sufficient motivation. Thank you and remember to go back up to the first slide has links where you can go do some more research and look for additional tutorials on this topic.