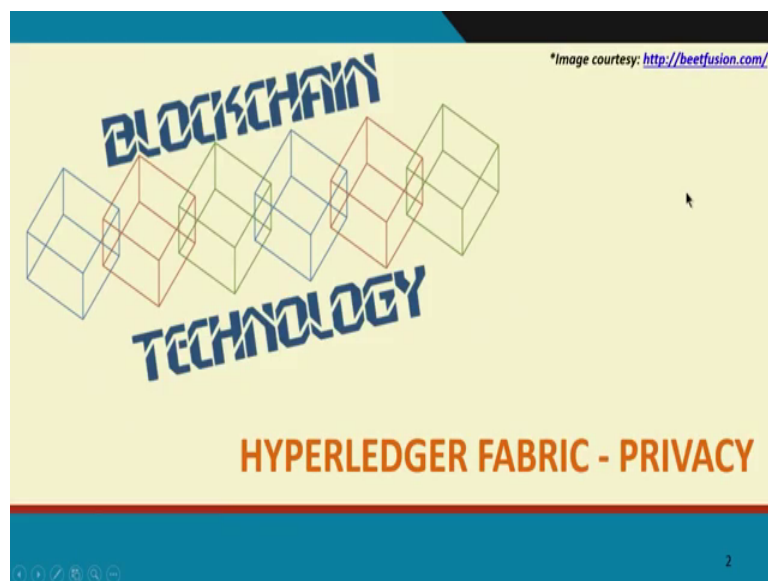


Blockchains Architecture, Design and Use Cases
Prof. Sandip Chakraborty
Prof. Praveen Jayachandran
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 46
Blockchain Security – III (Fabric SideDB)

Hello everyone. Welcome to the next lecture of our Blockchains Architecture, Design and Use Case this course.

(Refer Slide Time: 00:18)



So, in this lecture we going to look at a very important aspect of block chains. And it is also an aspect that has been seeing a lot of research activity lot of innovation which is really privacy. If you look at block chains as such right, it is about decentralization having multiple peers holding a replica of your data; thus, smart contracts are also replicated and decentralized. So, the notion of privacy is really at odds of this notion of replication and decentralization. So, how do you ensure that your transactions are seen only by a certain set of people? How do you ensure your data remains private and confidential?

So, the notion of privacy and the notion of replication and decentralization are really at odds. And there is a very interesting innovations that are coming up that allows you to achieve both notions right, both privacy as well as decentralization and replication across

multiple nodes in the network. So, in this lecture we look at what are some of those innovations that are all of these are very new concepts right within the last 5 years I would say. So, when we look at what we are storing on block chain. So, there is really multiple things we are storing, and the notions of privacy are also for that reason there are many notions of privacy. So, what do we really mean by privacy?

(Refer Slide Time: 01:45)

The slide is titled "Privacy in a Blockchain System" and features the NPTEL logo in the top right corner. It contains a bulleted list of privacy types and their support in blockchain platforms.

- **Transaction Data Privacy:** Transactional activity of an entity (profiling of the transactors)
- **State Data Privacy:** Chaincode / smart contract data (data that the smart contract alters)
- **Smart Contract Privacy:** Logic of the chaincode / smart contract (e.g., business logic)
- **User Privacy:** Anonymity and Unlinkability

Below the list, there are two additional points:

- None of these aspects of privacy are supported by permissionless blockchain platforms including Bitcoin and Ethereum, except pseudo-anonymity; Applications have to explicitly handle them
- Important aspect of permissioned blockchain platforms targeting enterprise applications

The slide has a blue footer with navigation icons and the number 3.

So, first of all block chain is going to store a set of transactions. So, there is a transaction privacy, oops. There is transaction privacy, which means that, I want to keep the inputs of a transaction, basically what is not contracts I am executing what are the input parameters, I want those to be kept private. So, that is the transaction data privacy. There is also the state debt data privacy. So, this is the internal state that is maintained by the smart contract and is recorded all that on to the ledger.

So, we want that data to also be private. I note that the transaction data is just the inputs to the function right. The state data is really what the function is actually computing and it could be more than just the input parameters that are sent in. The third thing is the contract privacy itself which is the logic of the chain code, which is really the code which is also going to be residing on the block chain we got to keep that private, right.

We want only expose that code to a certain set of authorized entities. And the final notion is you of that of user privacy, which is the fact that who is performing this transaction, who is endorsing a transaction, the user level credential details you want to keep

anonymous. And the other notion along with anonymity is that we also want unlinkability; which means that when one user performs let us say 100 transactions, no one should be able to say that these 100 transactions were really performed by one user right. Those 100 should appear as though they are they are really coming from independent entities.

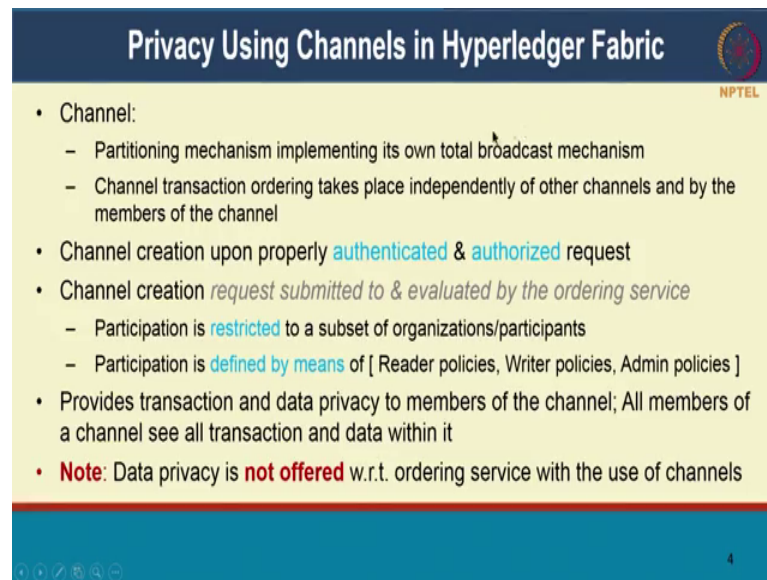
So, it is that is the notion of unlinkability. So, we want privacy at the level of transaction details, and in terms of state data, in terms of the logic of the chain code is the smart contract privacy and also user privacy. And none of these aspects of privacy are supported by the many permissionless block chain platforms. We have like including bitcoin and ethereum, they only provide the notion of pseudo anonymity; which is to say that I as a user I can assume a pseudonym. People will know that that pseudonym is performing transactions, but they really they cannot link the pseudonym with the original user.

But apart from that it does not provide any notion of transaction data or state data privacy at all. And it's left to the applications to explicitly handle these notions of privacy. And these notions of privacy are actually very important for many of the enterprise use cases. For instance, if I am storing healthcare records on block chain, the healthcare records are sensitive.

So, I want to make sure that the data I am storing is actually a private right. It does not matter whether I am storing it on block chain or otherwise, the intent from the use case or the industry is that that data has to remain private. And it can only be authorized for certain users on a need to know basis. Well, they say the other example is when a bank or banks are transacting all enterprises are exchanging money or assets between each other, they want to ensure that their user information is private, they do not want to know how much money is getting transferred between enterprises.

So, we want that notion of privacy and when the authorized entities should be able to see our critical information. So now, in this lecture we look at various ways in which some of these notions of privacy can be achieved, you know, in a block chain platform.

(Refer Slide Time: 05:21)



Privacy Using Channels in Hyperledger Fabric

- Channel:
 - Partitioning mechanism implementing its own total broadcast mechanism
 - Channel transaction ordering takes place independently of other channels and by the members of the channel
- Channel creation upon properly **authenticated & authorized** request
- Channel creation *request submitted to & evaluated by the ordering service*
 - Participation is **restricted** to a subset of organizations/participants
 - Participation is **defined by means** of [Reader policies, Writer policies, Admin policies]
- Provides transaction and data privacy to members of the channel; All members of a channel see all transaction and data within it
- **Note:** Data privacy is **not offered** w.r.t. ordering service with the use of channels

NPTEL

4

Hype energy of fabric as a as a permission block chain platform specifically targeting enterprise use cases has several constructs that all help provide different notions of privacy. So, then up to the application designer or the user to leverage the privacy in the right way, right. So, we look at some of those constructs, and we look at also some constructs that from other platforms along the way. So, the first notion that hype energy of fabric provides is the notion of channels, right.

So, channels are really a partitioning mechanism; whereby different subsets of entities in the network can all come together to form one channel. And all data within that channel are only available to the participants of that channel. So, participants are not part of the channel will not be able to see anything about the channel, they will not even know the existence of that channel. So, it is really a partitioning mechanism, and in some sense the each channel maintains it is own chain of blocks. So, the ordering itself is independent across channels. So, you could have different ordering services for each channel. So, each channels can have it is own ordering service. And those transactions are treated separate.

So, what are the things you cannot get is for instance, you cannot have a transaction that spans across channels. So, each transaction is restricted to a particular channel, and that also means that if you want to do something atomic which means that I want 2 pieces to transactions to happen on 2 channels, I cannot make that happen atomically using the

platform itself. So, the application will have to take care of that a atomicity. Then the coming to channel creation who can create channels only authenticated and authorized users will be able to create a channel.

Specifically, the channel creator who is creating the channel has to be a network administrator. The channel creation request is submitted and is evaluated by the ordering service the ordering service will check whether an authorized participant is trying to create the channel. And along with that that requests to create a channel you also specify how the participants are going to be right.

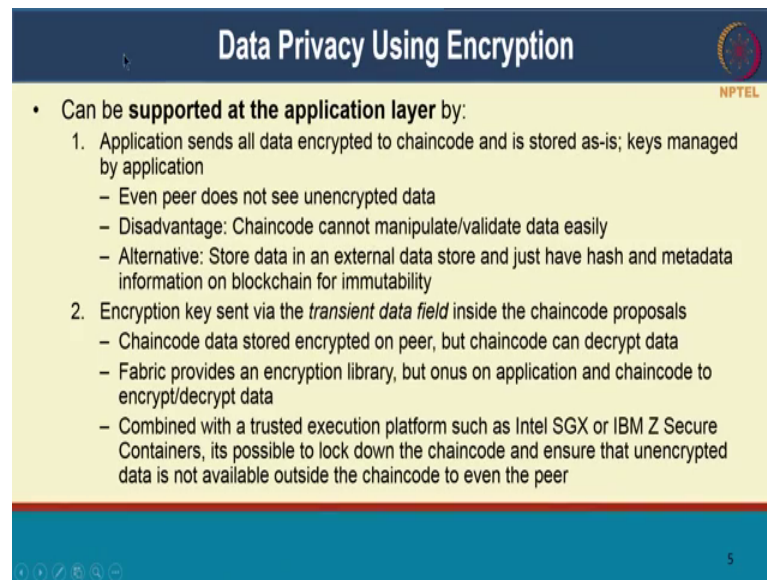
So, it is going to be restricted to a subset of the organizations participant. And it the participation is defined by 3 policies. One is who are going to be the readers. So, the readers in some sense you can think of it as the ones who are just going to be committing data. If they will be able to read data, all they are doing is committing transactions they are not doing anything else.

The writer policies is who can write data into this into this channel right so, there will be the writers. And the admin policies are the ones who are able to perform administrative functions. For instance,, only an admin will be able to install a change code or a sorry, instantiate a change code on a particular channel right. Likewise, only an administrator will be able to change the channel configuration parameters.

So, all those are administrative functions you there are of course, readers and writers onto that channel. And you can specify in a granular man or who can read who can write who has administrative privileges right. And so, this whole notion of channel in sometimes provides both transaction and data privacy to a subset of the members of the channel. So, all the members of the channel will have rights to see what transactions are there, what data is being stored in the smart contracts. But within a channel we do not have privacy in with this construct right. It just allows you to partition transactions and data in the network. And all members of a channel will see all the data within it. Note that the data privacy is not offered with respect to the ordering service.

So, the ordering service for a particular channel will see the data within the channel, and with just using channels you cannot get data privacy with respect to orders. So, others will see the data, and there are other constructs that have that will help you even of course, how to state that from the orders.

(Refer Slide Time: 09:29)



The slide is titled "Data Privacy Using Encryption" and features the NPTEL logo in the top right corner. The content is as follows:

- Can be **supported at the application layer** by:
 1. Application sends all data encrypted to chaincode and is stored as-is; keys managed by application
 - Even peer does not see unencrypted data
 - Disadvantage: Chaincode cannot manipulate/validate data easily
 - Alternative: Store data in an external data store and just have hash and metadata information on blockchain for immutability
 2. Encryption key sent via the *transient data field* inside the chaincode proposals
 - Chaincode data stored encrypted on peer, but chaincode can decrypt data
 - Fabric provides an encryption library, but onus on application and chaincode to encrypt/decrypt data
 - Combined with a trusted execution platform such as Intel SGX or IBM Z Secure Containers, its possible to lock down the chaincode and ensure that unencrypted data is not available outside the chaincode to even the peer

So, the second construct is of course, you can use encryption right. So, this is a well known technique for decades using encryption for ensuring privacy. So, you can encrypt the data, and share the key only with authorized entities, and that way only the authorized entities will be able to decrypt the data and see the see the original content. Everyone else will see just the encrypted data, and with that they cannot figure out what the original text is.

So, there is a well-known security primitive just using encryption. But then the application will have to do this encryption. At least in most of the block chain use cases otherwise, you are you are you do not want to share the key on the block chain that way then everyone will get the key, right. So, then you losing your privacy right. So, once you encrypt the data at the application layer, the application now has to maintain these keys, now even the peer will not see the unencrypted data all right. But then if the chain code is also only going to get the encrypted data and all it will do is to store that data, and I will be able to retrieve that encrypted data.

Now, the disadvantage is that it is going to be very hard to manipulate or validate what that data is. For instance, I cannot do arithmetic operations I cannot do conditional operations. Yes, there are constructs like homomorphic encryption, that do allow you to perform certain arithmetic certain functions on that encrypted data itself, but it is it is not it comes at a performance penalty and there are certain disadvantages to doing right.

Alternatively, you could store a private data off chain, all right you can store it in an external data store ensure that the right parties have availability of the data. And just have the hash and metadata information stored on the block chain. So, that will give you the immutability. So, if someone goes and changes the external data store the data in the external data store, anyone can actually check the hash on the block chain, and determine that something is wrong I think the hash does not match any more.

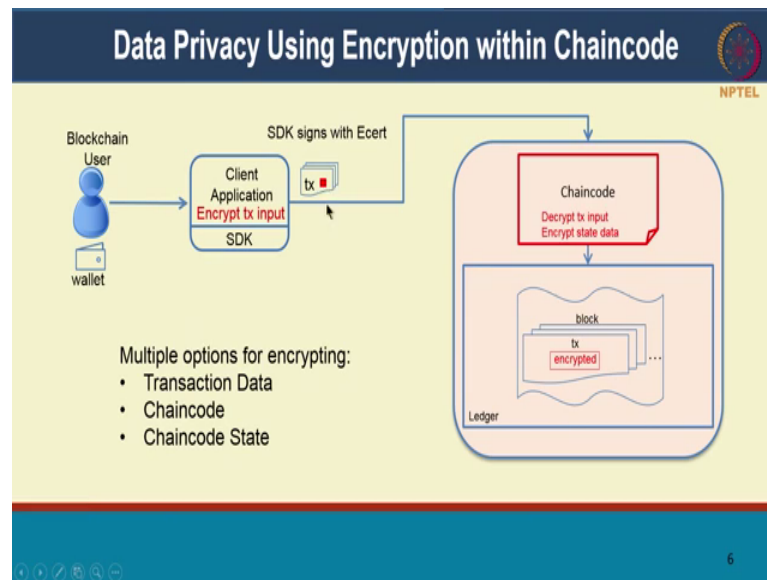
So, you know that at ease you can detect that someone has changed the data have tampered with the data. So, this gives you mutability, but does not give you all the other advantages of smart contracts of being able to modify the data in a decentralized fashion on a block chain network. So, that is that is about application layer encryption. Great, the other thing you can do is with fabric, you can actually send the encryption key to the chain code. So now, you want to let the chain code also see the decrypted content. So, how does this work right so, our chain codes have this field called transient data right. So, this is a data field in your chain code parameter. So, when you are invoking a chain code, you can also add a transient field, and send some secret information in the transient field right.

So, when you send that information the chain code will be able to retrieve this transient field information, but that information is not included in the transaction. So, if you see the transaction, it will have the chain code that is invoked it will of all the parameters, but the transaction itself will not have the transient field. That is why it is called transient ok. Now what the chain code can do is you can it can use that transient key and store data encrypted on the peer.

So, that way the peer will not really have the decrypted content, but the chain code itself can work with the decrypted content, can encrypt and decrypt using that key right. Now this along with leveraging let us say our trusted execution platforms such as intel SGX or IBMs esecure containers, it is possible to isolate the chain code container from the pure.

So, that way the chain code will execute in a trusted environment. And even if the chain code is handling the encrypted key and as decrypting the content. The peer will not get to see that content at all. The data when it gets in the peer will in encrypted form and will be stored in encrypted form so, it is possible to do that. So, this picture just gives you the flow of how that works right.

(Refer Slide Time: 13:29)



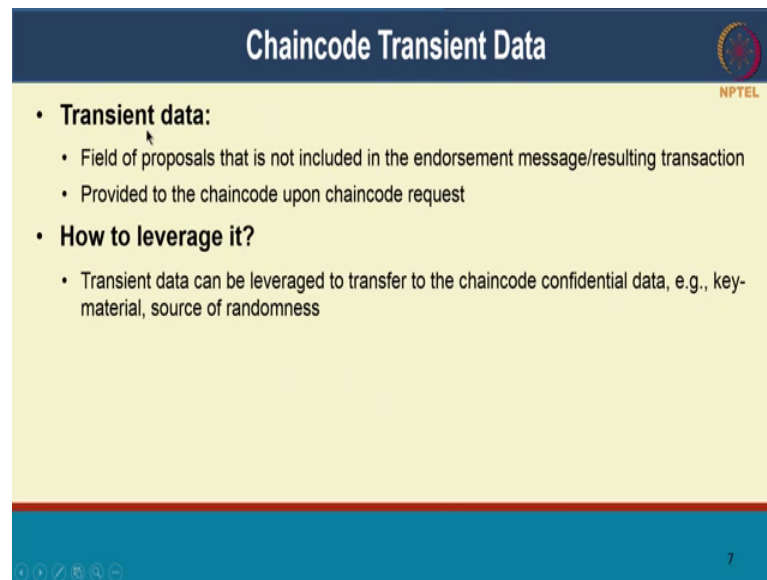
A block chain user is going to use a client application to invoke a transaction on the particular chain code. So, what you will do is they are going to encrypt the transaction input parameters. So, all the input parameters it is going to go into the chain code, they are going to encrypt it. And of course, the transaction is going to be signed with the enrollment certificate.

So, this enrollment the key there is different from the encryption key. So, the what the key you use for signing is not the same as the key you use for encryption. And you for different data parameters you can actually use different keys. And what you are going to do is send that, the send the keys to the chain code through the transient field. And now the chain code can actually decrypt the transaction input it can perform certain whatever functions it means to perform.

It will then encrypt the state data and then store it onto the block chain. So, when it gets to the pure and what you will see on the block itself on the block chain will only be encrypted data. So, overall this whole notion can be used to encrypt the transaction parameters, it can you be used for used to encrypt the chain code logic itself, just the code itself is going to be part of our deploy transaction right.

So, that can also be encrypted, and the chain code state can also be encrypted what you are storing on the block chain right. So, all of these are now are now possible with hyper larger fabric.

(Refer Slide Time: 14:59)



The slide is titled "Chaincode Transient Data" and features the NPTEL logo in the top right corner. The content is organized into two main sections: "Transient data:" and "How to leverage it?".

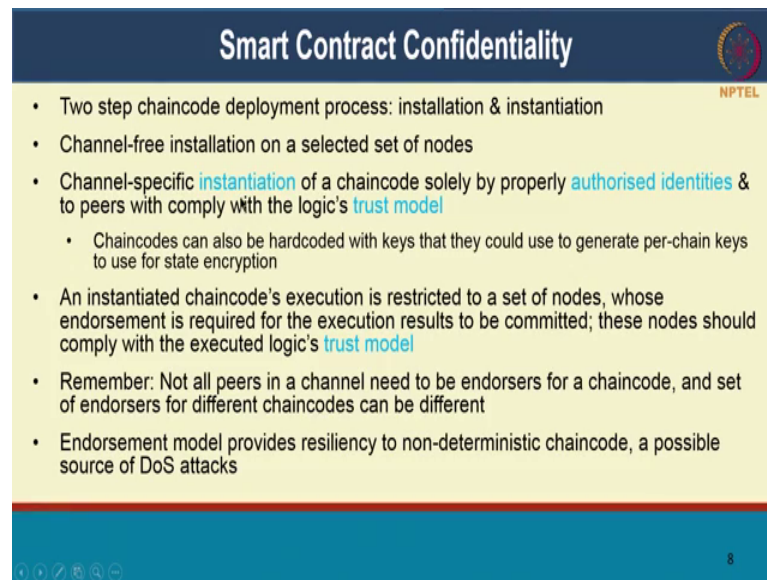
- **Transient data:**
 - Field of proposals that is not included in the endorsement message/resulting transaction
 - Provided to the chaincode upon chaincode request
- **How to leverage it?**
 - Transient data can be leveraged to transfer to the chaincode confidential data, e.g., key-material, source of randomness

At the bottom of the slide, there are navigation icons and the number "7".

So, the transient detail data itself it is it is not included in the endorsement method it is not including the inference in the transaction. And it is provided to the chain code upon request right. And how do you leverage it can be used for sharing confidential data like key material; it can also be used as a source of randomness right. So, typically chain codes you do want to want it to execute you know in a in a deterministic fashion, how many of our peers executed right.

So, even if multiple peers are executing the same code, you want that to be the execution to be deterministic. So, you do not allow any you do not want any kind of randomness to come in, but what you can do is you can send in a seed right. So, that all of these peers will generate the same random number, but will be random for each transaction all right. So, that source of randomness can also be sent to the chain code using this transient field.

(Refer Slide Time: 15:58)



Smart Contract Confidentiality

- Two step chaincode deployment process: installation & instantiation
- Channel-free installation on a selected set of nodes
- Channel-specific **instantiation** of a chaincode solely by properly **authorised identities** & to peers with comply with the logic's **trust model**
 - Chaincodes can also be hardcoded with keys that they could use to generate per-chain keys to use for state encryption
- An instantiated chaincode's execution is restricted to a set of nodes, whose endorsement is required for the execution results to be committed; these nodes should comply with the executed logic's **trust model**
- Remember: Not all peers in a channel need to be endorsers for a chaincode, and set of endorsers for different chaincodes can be different
- Endorsement model provides resiliency to non-deterministic chaincode, a possible source of DoS attacks

So, the next notion so, we looked at how you love channels can help transaction data and prior transaction and data privacy, now we looked at encryption, and how the transient field notion and chain code scan actually give you different notions of encryption and privacy. The third thing is how do you ensure that the code itself is going to be remain confidential right.

So now there are 2 parts to this the first is there is it is important to understand the deployment process and fabric. So, how do you deploy a chain code. There are 2 steps to it one is the installation should step, and the second is instantiation. So, the installation is really about being able to run hack the logic in a particular peer. So, this is independent of channels on the peers where the chain code is installed, the peer will have that logic of the chain code all right.

So, this is just based on the peers. So, which peers have to run, this code they will have the chain code so, that is the installation right. So, the next part is a channel specific instance. So, based on for each channel on which you want to instantiate a chain code, you will call an instantiation transaction. And at that point you will say who are the authorized peers, who need to execute this transaction and may act as endorsers for this transaction.

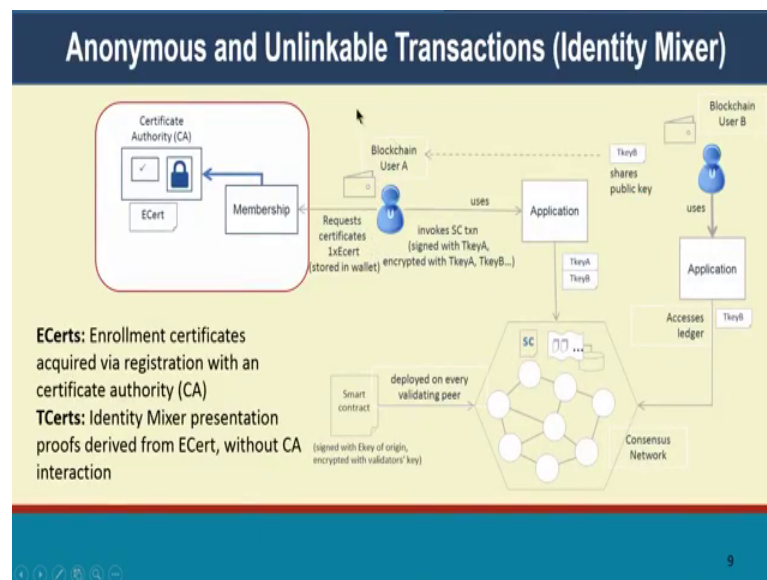
So, that is the second notion. So, the fact that endorsers can be a subset of all the entities in the channel, means that the logic only needs to reside with the endorsers. So, let us say

my channel has 10 participants, I can say only 5 of these 10 participants a specific set of 5 participants are going to be endorsers right. So, only those 5 will then have the chain code logic the remaining 5 will not have the chain code logic at all, but will see the transaction data will see this chain codes data, but they will not see they see the logic or the code of the chain code itself.

So, when the endorsers will have the code and within those endorsers of course, you can have the endorsement policy. So, you can say for a particular transaction 3 out of 5 and ourselves have to sign right, but only the 5 people will have the code itself. And so, this ensures a smart contract confidentiality. So, only the peers for a particular channel that need to execute the transaction who are good actors endorsers will or can hold the smart contract code right. And the endorsement model provides resilient resiliency to non-deterministic chain code.

So, given that multiple endorsers will endorse a particular chain code, you can ensure that that code is deterministic right, ok.

(Refer Slide Time: 18:44)



Now, coming to the next notion right; so, you are coming to the notion of user privacy no all right. So, right now there is there is a identity mixer which is which is a really an open source project from a IBM, it came from the crypto world. But it is now being adopted by hyper ledger of fabric itself. It is going to come in the next release, version 1.2 of fabric. The development is already underway; I think we briefly talked about this in an

earlier lecture. But identity mixer gives you a very strong guarantees on anonymity and unlink ability.

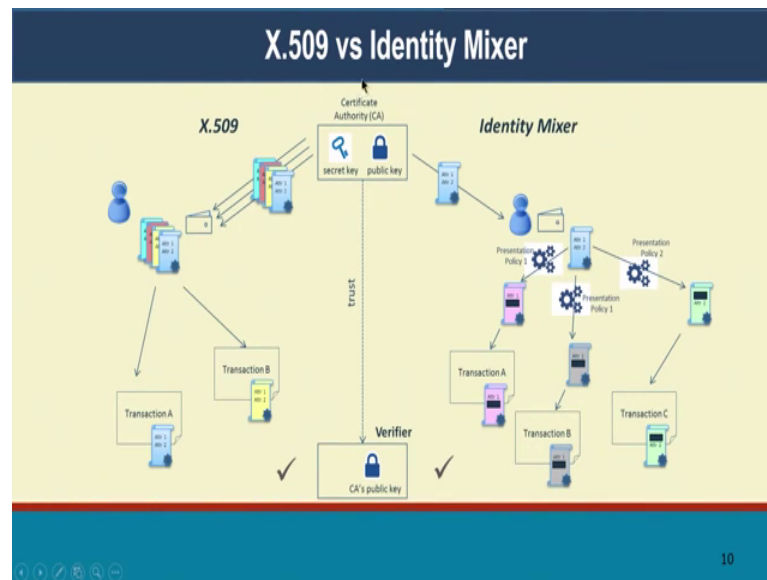
So, let us look at how at least at our high level how this works, right. So, there is a block chain user so, as before they have to obtain certificate from a certificate authority. So, there is a membership service provider that interacts with the certificate authority to issue certificates. And there is enrollment certificate as before right. But now the user is going to when they invoke smart contract transactions. So, when they invoke transactions, previously we had mentioned that the transaction is going to be signed by the enrollment certificate. Instead, what is possible with identity mixer is that the user can generate per transaction certificates.

So, these are specific one time use certificates, that the user can use for different each transaction. And they can also have other one time use keys that are generated suggest for encrypting data, like I talked about before right. So, you can generate keys for signing, you can generate keys for encryption. And all of these the it is possible that the user can actually prove that they were all these keys were derived from their enrollment certificate, right. So, that is really a unique ability; where the user can generate multiple keys from an original ECert, but they can also prove to an auditor that these were all generated by that ECert.

So, the auditor can come in and see that these were all performed by these transactions were all performed by the same user, but no one else in the system will know that they were performed by the by one user right. So, that is the difference with ECert, and the TCert the transactions certificate will be one time certificates that the user can generate to perform transactions on block chain. So now, because all of the TCert look different right, even if the user is performing 10 transactions, they can all have n different keys. So, it appear as though these word up these were performed by 10 different users.

So, that way you achieve unlink ability right.

(Refer Slide Time: 21:30)



So, let us just compare the previous X.509 certificates that I talked about with identity mixer. So, think of identity mixer identity mixer is really another fabric CA and our fabric or membership service provider implementation; which is distinct from the fabric CA that we talked about earlier. The previous fabric CA was based on X.509 certificates and well see how they are different right. So, on the left is how X.509 roughly works right. So, there is certificate authority, that is issue going to issue a certificate to a to a user. And this certificate can have multiple attributes.

So, we have called out attribute 1 and attribute 2. So now, the user has one certificate with 2 attributes. So, they have these 2 attribute 1 and attribute 2, and whenever they perform a transaction, they are going to sign with that certificate. So, any anyone else who is seeing this transaction will see both their attributes. So, that way they will know that this was performed by this user. So, there is no anonymity there, and there is also no unlink ability. So, between these 2 transactions, I know that these were both performed by the same user. So, in this model in the previous fabric C there was no anonymity or unlink ability.

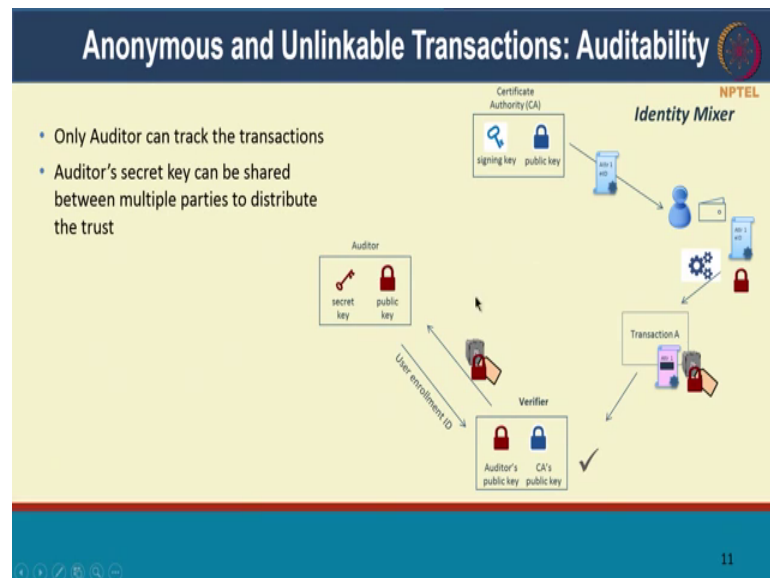
So, let us look at identity mixer and what it does right. So, the identity makes her the CA for identity mixer issues like before an enrollment certificate. It has like before 2 attributes attribute 1 and attribute 2. Now what user can do? Independent of having to

talk to the certificate authority, they can have a presentation policy. So, there can be new policies.

Based on policy 1, it can choose to just disclose attribute 1. So, it generates a transaction certificate that has just attribute 1, and policy 2 based on policy 2 may be this is this can be even a different channel altogether, where they are the users performing transactions. It can be may be a different chain code, or just simply a different transaction within the same chain code. For transaction A I want to disclose attribute 1 for transaction B, I want to disclose attribute 2.

So, if you look at this attribute 1 is hidden here and only attribute 2 is shown. And in this transaction only attribute 1 is shown. So, that is possible so, you can have distinct transaction certificates, each exposing different attributes that a user has. Anyone you can present that and you can use that to sign transactions. So, these transactions will look completely different to an end user. So, this way the ECert is also hidden. So, you get complete anonymity, and you also get unlink ability across transactions.

(Refer Slide Time: 24:13)



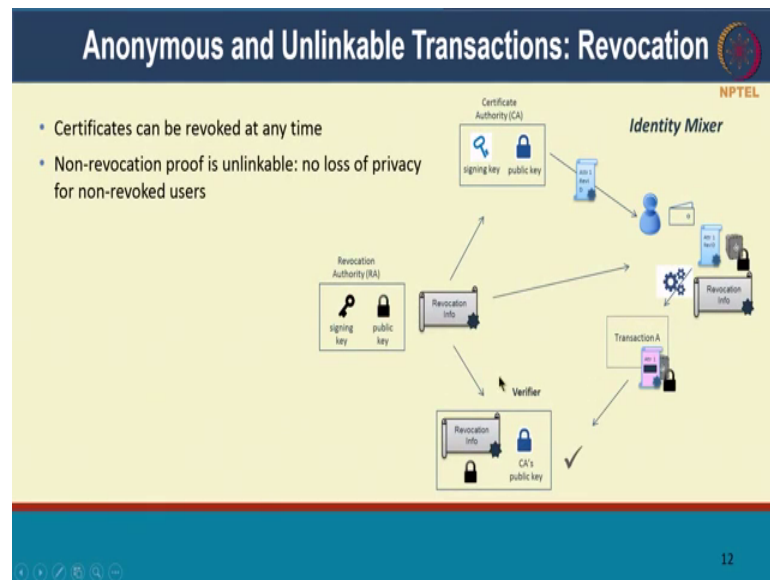
So, the next thing that you can perform with these anonymous and unlink able transactions, I just sent edited before these auditability. So, what is possible to do is once signed these transactions, I can also I can actually prove to an auditor. So, let us say auditor has a particular secret key. I can prove I can basically expose this information only to an auditor. And the auditor will be able to verify this transaction to say, that this

attribute indeed belongs to this user, and this is a valid certificate, the auditor will be able to determine that. And not only that it is possible for it is not that one auditor will see all the data, it is possible to have per data element auditor or per attribute audit.

So, I can say only for this attribute this person is the auditor, for some other attribute it can be a different auditor altogether. So, that is also a very unique capability that we have in hyper ledger fabric. And it really gives you a very cool very fine grained control on exactly what data elements you want to share with whom including with auditors. So, you can say this data element, I will only share with this auditor and not with someone else right. So, that is also possible. And when I say auditor it need not be just one entity that holds that key. It is possible to share that key with multiple or dispute that key with multiple parties.

So, only if all of them all of those parties come together will be will they be able to audit this thing. So, those are known capabilities and cryptography, yes. So, that is the unique notion of audit ability that hyper ledger fabric can give you.

(Refer Slide Time: 25:58)

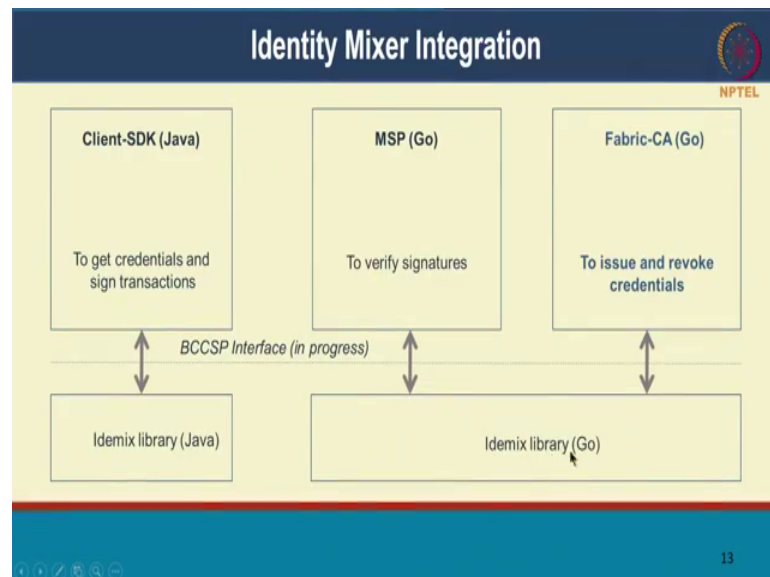


And of course, this certificate revocation; so, this happens this is an important functionality provided by the certificate authority. So, it is going to maintain a certificate revocation list of these set of users user certificates that have been revoked, so, with identity mixer. So, there is a separate revocation authority. And what is possible is that the when some so, basically whenever someone is verifying a transaction, they are going

to go back to the revocation, or the vocation list and check whether this certificate is still valid or whether it is if it is been revoked.

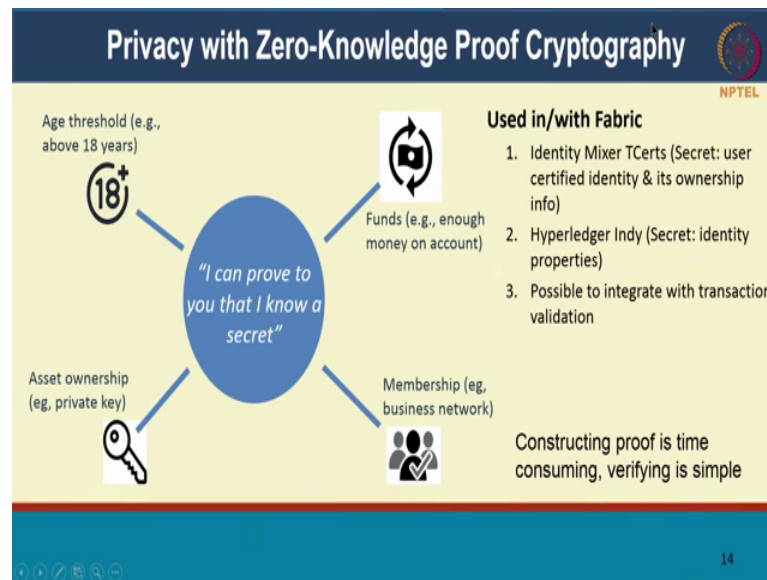
Let so, to do that it is so possible to do that in a privacy preserving man. So, all non-revoked users there will be more loss of privacy because of having this non revocation proof; to show that this certificate has not been revoked, can also be done in a privacy preserving manner. So, right now the implementation for the integration of identity mixer with hyper ledger fabric is going on, and should be out in probably a month or so. I think it is it is already in the road map for fabric 1.2 version, right.

(Refer Slide Time: 27:11)



So, so there are libraries written partly in go partly in java. And this is really new block chain crypto service provider interface that is being written. So, there is our client SDK, this going to be a new fabric C and of course, the membership service provider implementation, that provides these functionalities, ok.

(Refer Slide Time: 27:33)



So now I am getting to another very interesting cryptographic notion, which is 0 knowledge proof. So, 0 knowledge proof are really almost they seem like magic, right. So, what they are is that I want to be able to prove that I hold a particular secret or that I have some particular private information. And without revealing that information to you, I want to prove to you that I actually have that private information. For instance, one of the things I want to do or one of the common cited examples is I have a drivers license, the drivers license has my age on it. Typically, when I go somewhere and present my driver's license. You will see all the information on it right, you will see my date of birth my address everything right. But let us say I want to only prove to you that I am greater than 18 years of age right. I do not want to reveal to you all the other details of my driver's license. Can I do that?

So, there are ways with cryptography; where without revealing my driver's license or my date of birth, I can prove to you that I am greater than 18 years of age all right. That is pretty cool, isn't it? So, that is really the 0 knowledge. So, you will once you have the proof you can be guaranteed that I am actually 18 years of age or older, but I have not revealed you really revealed, I have given you 0 knowledge about my private information. I have not revealed my date of birth at all and there is no way for you to guess that ok. Another very important use case is my funds right. For instance, I do not want to reveal my bank balance, but I want to prove to you that I have enough money in my bank account in order to make a payment. Let us say I am go to pay you 100 dollars,

all I need to prove to you that I is that my bank balance is more than 100. I do not have to reveal my bank balance to you for that right.

So, that is again a very powerful notion. There is also our proofs about membership right. So, I want to prove to you that I am a legitimate user of the network, I am permissioned user, but I do not want reveal my identity. So, that is also another place where there is 0 knowledge is used; where I do not reveal my identity, but I prove to you that I have I am authorized for access ok. And of course, asset ownership I want to prove to you that I own an asset without revealing to you what that asset is. So, there are multiple of these use cases for 0 knowledge proof. And this is actually in cryptography this started way back in the 1980's over a period of 30 years, it is actually matured to a good extent, and now with block chain coming into play and privacy being an important aspect of applications being built on top of block chain. People are asking for these privacy notions, 0 knowledge proof have seen a resurgence over the last maybe 3-4 years, there is a lot of work going on in 0 knowledge proof.

And let me give you a few examples all right. Identity mixer with the transaction certificates actually uses 0 knowledge proof; where the user can prove to you that this t sat was in fact generated from a particular ECert, without revealing that ECert to you all right. So, that is in the identity space. Likewise, hyper ledger in D which also focuses on the identity can prove to you certain claims about your identity. So, claims are issued by different people, and I can prove to you that I have a particular claim without revealing that claim to you. And it is also possible to integrated with transaction validation. So, this is with the state verification whether the inputs and outputs of an transaction it is possible to validate them. So, I will come to that in the in the next slide. And an important property of these 0 knowledge proof is that the proof is actually time consuming to create. And even with the private information, it is going to be very hard it is actually takes it is computationally intensive for the owner of the information to construct the proof, but verifying the proof is very simple.

So, you can really the verification can be done by a large set of independent parties. They can all verify it very quickly, but construction of the proof is very, very difficult. So, that is the cryptographic property that that they have. And of course, if I do not have the private information, it is impossible for me to match a proof and to make you believe that I actually have that information. So, without the private information I cannot fool you.

So, you can be guaranteed that it is impossible for me to generate the proof without the private data ok.

(Refer Slide Time: 32:15)

The slide is titled "ZeroCash: UTXO Ownership Model with Privacy" and includes the NPTEL logo. It contains the following text:

- Zero Knowledge Proof for:
 - Full anonymity based on ZK-SNARKS (not just pseudoanonymity); prove you have private key
 - Concealing asset value transferred (inputs and outputs values in transaction); prove transaction is not double spending or generating coins
 - Conceal UTXO graph
- ZeroCoin cryptocurrency as extension to Bitcoin
- Improved in 2013 as ZeroCash, with 98% smaller proof sizes
- Time taken and size of proofs considered a concern
- Integrated with Ethereum, Quorum (variant of Ethereum for permissioned networks); ability to verify ZK-SNARKS on-chain

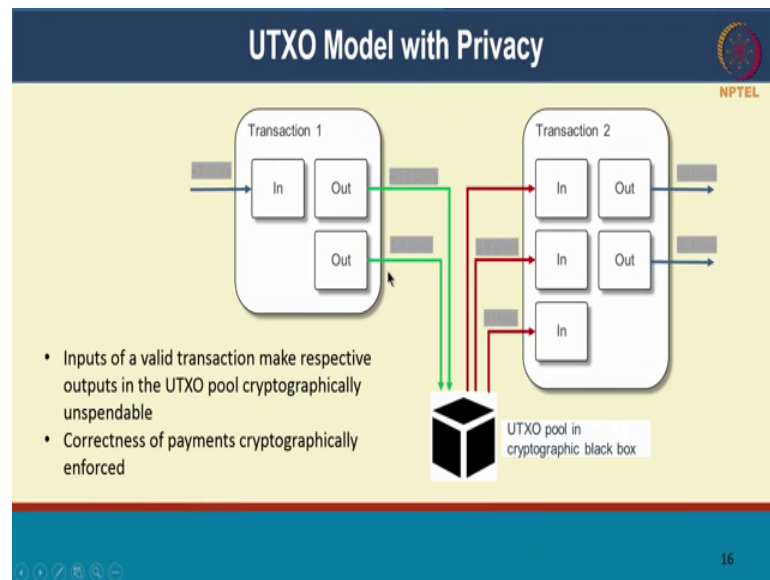
At the bottom right of the slide, the number "15" is visible.

So, let us look at a few other examples; where 0 knowledge proof are employed in a very nice way ok. So, 0 cash is a very nice example. So, they what they have is 0 knowledge proof for providing full anonymity of users right. So, remember that bitcoin and ethereum only provide pseudo anonymity right. They do not give you a full on anonymity alright. So, what they can do with 0 knowledge proof is that they can prove that they can ensure that the user identity is never revealed when someone looks at a transaction, right. They can also conceal the asset value. So, in bitcoin for instance everyone who is looking at the network can see by just looking at a particular transaction they know exactly how many bitcoins got transferred from one bitcoin address to another.

So, that bitcoin value is not private, but what 0 cash allows you to do is to conceal that value right also though the what if the way now we will talk about how does that in the in the next slide. And the other thing we can also do is conceal the UTXO graph. So, basically the list of unspent transactions that can also be kept private. So, 0-coin was a it introduced as the cryptocurrency has an extension to bitcoin. So, they actually forked from bitcoin and they created a new car currency cryptocurrency based on 0 cash and is called 0 coin. And in 2013 they actually showed that a 98 percent smaller proof sizes.

So, they dramatically improved the complexity of these proofs. The size of the proof became very small. And the time taken, but the time taken is still a concern right. So, there is still this still takes significant amount of time and computational effort to construct these proofs. So, that is still a concern and it is act it is an active area of research. And this whole 0-coin notion has been integrated now with etherium and quorum. So, it is possible in etherium to conceal the transaction information right. So, who is transferring how many ethers to someone else. And the ability to verify these 0 knowledge ZK-SNARKS as they are called on chain is now inbuilt into etherium and quorum. So, we look at etherium in quorum later in this lecture, but this has been incorporated there.

(Refer Slide Time: 34:46)



So, let us quickly look at how this works right. So, in any bitcoin transaction or a etherium transaction basically any UTXO there are a certain set of input inputs and a certain set of outputs. And what we are doing when we win this chain when we change these transactions together; is that every transaction needs to prove that the inputs are all unspent so, if they have to be unspent outputs of other transactions. So, there are these links in some sense, and what we want to hide are the following right. We want to hide the amount of value that is getting transferred the in and out all right, how much is the value of the transaction. We want to hide the identity. So, I talked about hiding the identity, and we also want to hide the UTXO pool.

Now, if you look at the validation logic or the prevention of double spending, what we want to ensure is just that the outputs are not more than the inputs. In some sense we are not generating new assets or new money out of this right the outputs have to match the inputs. All we have to prove is that property, right we do not have to reveal exactly what the inputs are what the outputs are. So, those values can be hidden, but I prove to you that outputs are not more than the inputs. So, that proof is sufficient for you to ensure that there is no double spending, right.

So, that is provided through a 0 knowledge proof. So, someone who is submitting this transaction has all of these inputs and outputs of the obfuscated. They along with that they submit a proof saying outputs are not more than inputs right, anyone in the network all the peers let us say bitcoin or ethereum peers, they can all verify that outputs are not more than inputs. So, it is a valid transaction and they can admit it. And what is also great is that the set of unspent transactions can also be kept private. So, no one at any point of time even the peers will not know what the set of unspent transactions are. But what they can verify his membership right, I can verify that this particular input is a non-spent transaction at this point. It is part of this private set in some sense that set is kept private even the peers do not know it, but I can prove that I belong to that set.

So, isn't that cool that is really a very cool property, this very involved mathematics in this all of which I myself do not understand. But the properties are very, very useful in many enterprise applications right, and these are becoming reality today. So, we are actually starting to use these very complex cryptographic primitives for providing privacy in block chain platforms, ok.

(Refer Slide Time: 37:32)



The slide is titled "Fun Reading" and features the NPTEL logo in the top right corner. It contains a list of four items, each with a link to additional resources:

- TEDx talk, Using bitcoin blockchain to detect fraud: <https://www.youtube.com/watch?v=507wn9VcSAE> (linking transactions on bitcoin blockchain)
- Identity Mixer, website: https://www.zurich.ibm.com/identity_mixer/
 - Overview: https://www.zurich.ibm.com/pdf/csc/Identity_Mixer_Nov_2015.pdf
 - Github: <https://github.com/IBM-Cloud/idemix-issuer-verifier>
- Zero knowledge proof, Wikipedia: https://en.wikipedia.org/wiki/Zero-knowledge_proof
- Zerocash project: <http://zerocash-project.org/>
 - Research paper in IEEE Symposium on Security and Privacy, 2014: <http://zerocash-project.org/media/pdf/zerocash-oakland2014.pdf>

The slide also includes a navigation bar at the bottom with icons for back, forward, and search, and the number 17 in the bottom right corner.

With that we have come to the end of this lecture. We will look at a few more at least one more privacy primitive in hyper ledger fabric in the next lecture. But in the meanwhile there are a few interesting reading right, there is a lot of witnessing work going on in the space as I said, this is a great topic for research if you are looking for like athes is topic.

So, let us go over what some of these things right. So, there is a nice ted talk TEDx talk on using the bitcoin block chain to detect fraud. So, this is really from like a lawyer in the US; where they actually monitored transactions on the blog on the on the bitcoin block chain to detect people who are doing some fraudulent transactions. So, I will not go into the details, but it really tells you 2 things right.

One is how bitcoin actually does not give you the privacy, but at the same time because of the fact that block chain immutably records all transactions that have happened, it is actually possible to go back in the past and know exactly what happened who did what. So, the provenance is great for detecting fraud, but at the same time there is privacy concerns in the bitcoin block chain.

So, that is a very interesting about 20 minutes I think. It is a very interesting ted talk. Identity mixer; so, you can find out more details of if you are into security into cryptography, I would encourage you to take a look. So, this is completely open source identity mixer itself is open source there is a lot of material for you to do look into the details. There is a nice overview set of the overview article. And there is a link to the

GitHub itself including documentation. And if you are interested in 0 knowledge proves in particular the Wikipedia is a great place to start it gives you a nice overview of concepts. And the 0 cash project itself is a very interesting project. So, I would encourage you to read that they have a research paper in the IEEE symposium on security and privacy which is also a very good read. So, I had encourage you to dig deeper into this topic if that interests you. And with that well get to the next lecture, we will we look at one more construct for privacy.

Thank you.