CL Signatures for Anonymous Credentials

Will Abramson · Follow
Published in Good Audience
6 min read · Jan 14, 2019

Listen Share

A CL Signature is a signature scheme developed by Jan Camenisch and Anna Lysyanskaya. This scheme has some properties that make it ideal for use in an anonymous credential system and is, in fact, the scheme that Sovrin, and I am sure others, currently use. In this post, I will try to synthesise my current understanding of this scheme, including a look at how Sovrin uses it in practice.

Note: this is just my current understanding, please correct me if you notice any errors or miscommunicated concepts. For the more adventurous, these are the three papers I recommend reading: 1, 2, 3. It is also worth taking a look at the idemix protocol specification, which was designed by IBM Zurich as a specification for an anonymous credential system.

The Properties of CL Signatures

CL signatures enable the issuance of a signature on a committed value, a blinded signature. Blinded signatures lots of different uses, most famous probably being in untraceable electronic cash designs first introduced by David Chaum in this paper. The idea is that the receiver of the signature can commit to some secret information that only they know, the issuer then signs this committed value and presents the signature to the receiver.

This is key in issuing an anonymous credential. When creating a signature on an anonymous credential it is required that the recipient of the credential commit some secret information to this credential. The commitment to this secret information is then signed by the issuer and forms part of the credential. It is used

by the recipient of the credential to prove that the credential was indeed issued to them and no one else.

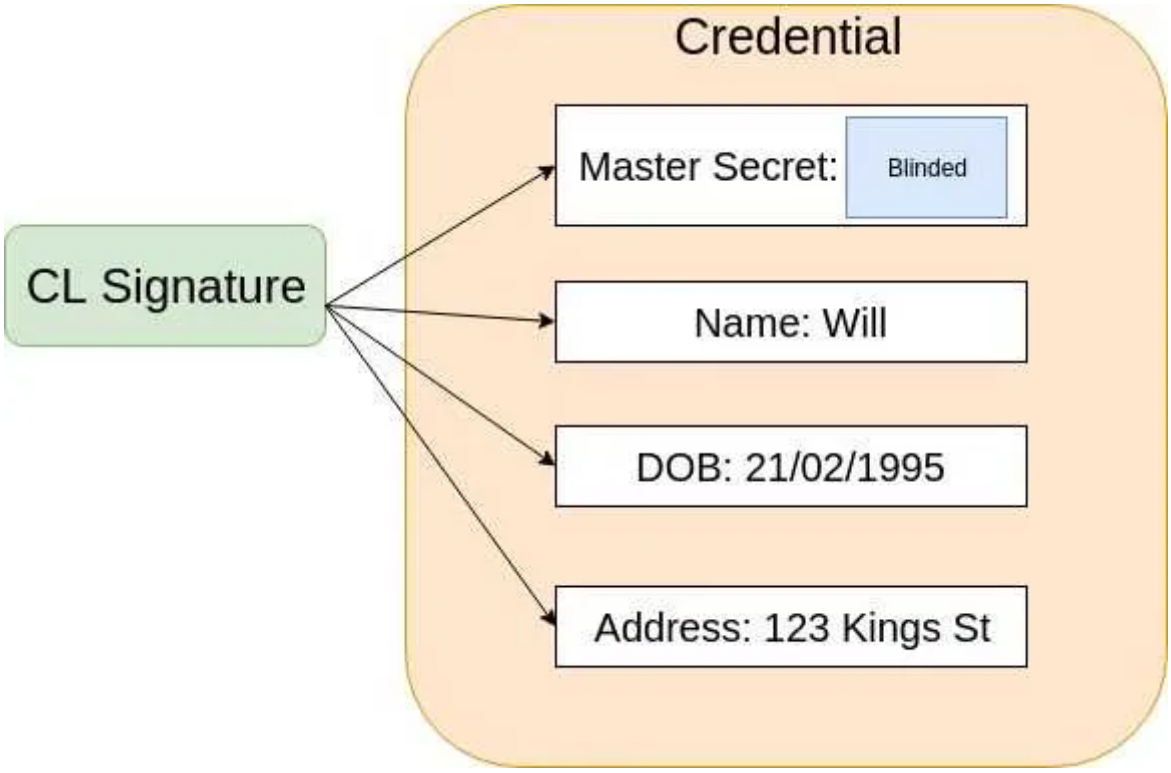
This leads to the second property of CL Signatures. The ability to prove knowledge of a signature on a committed value. To prove the credential I am presenting is indeed mine to present I must prove I have knowledge of the signature on the committed secret information included when the credential was issued. This is an area I could use a little deeper understanding, but it seems logical that one can only produce a ZK proof of knowledge of a signature on a committed value if they indeed have knowledge of the secret value the signature was created on.

The idea behind an anonymous credential system is that an entity interacts with other entities using different unique pseudonyms for each connection (or at least for some connections). However, when a credential is issued it needs to be tied to the entity as a whole rather than to the pseudonym with which the entity is known by the issuing party. CL signatures allow for control of credentials to be tied to a master secret known only by the entity, meaning credentials issued across one connection can be proved across another without loss of privacy through the linkability of pseudonyms. As a credential issuer only ever sees a commitment to the master secret, which can be randomised, they learn no information about the secret. A possible downside to these systems is that if a master secret were to be compromised then all credentials could potentially be stolen. Research is currently exploring making these systems more robust through the social recovery of keys and other mechanisms.

A credential is made up of more than just who is in control of it. A credential like your driving licence has your name, DoB, address etc. To make this signature scheme usable for credentials the scheme needs to enable signing blocks of messages. In an anonymous credential, each message block represents an attribute. Rather than amalgamating these message blocks into a single message using a collision-resistant hash function, CL signatures enable a useful feature — effectively signing each of these message blocks in turn. This means that any subset of these attributes can be presented along with a valid signature, allowing for much more fine-grained, privacy-preserving credential presentations than in the physical world.

Remember one of these attributes must be the blinded secret enabling the receiver of the credential to attest being in control of it. This means that a CL signature on an

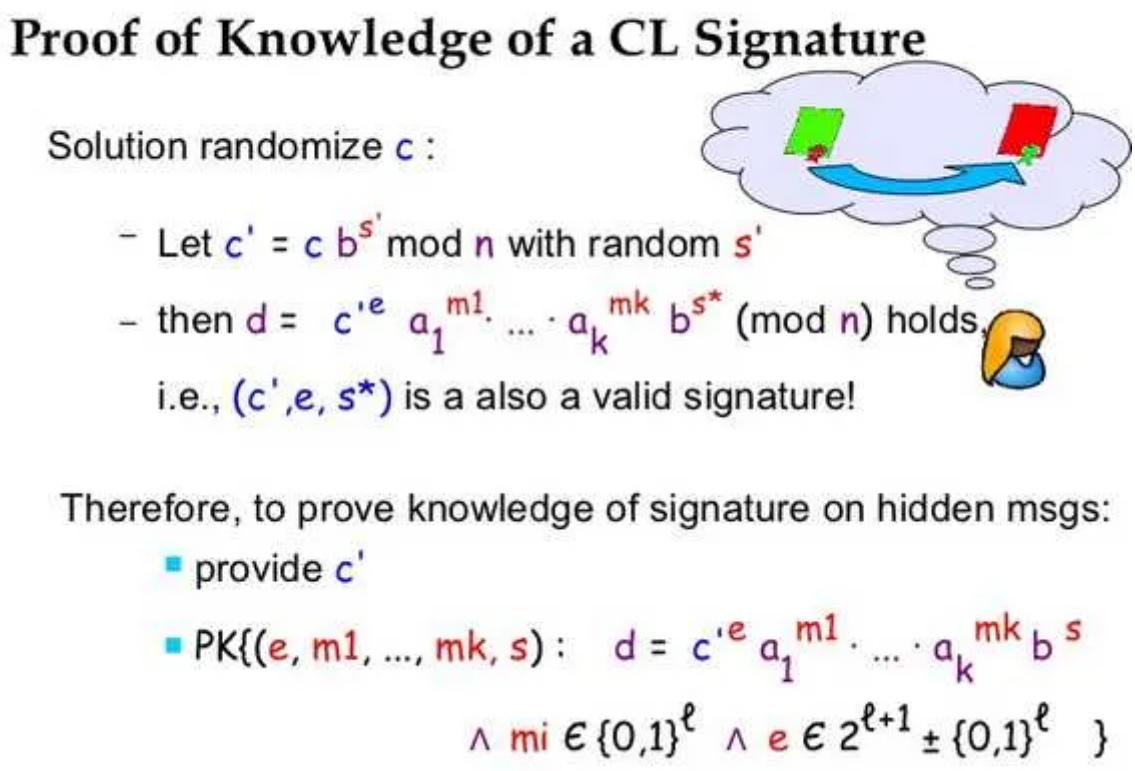
anonymous credential is most often a partially blinded signature on the blinded secret value and the other unblinded attributes of the credential.



Another property of CL signatures that makes them useful in a credential system is the ability to combine them. An entity can create a single proof containing attributes from multiple credentials. A credential proof is made up of predicates (evaluating to true or false), to combine attributes from multiple credentials the prover must submit proof of the CL signature on each credential. It is often restricted so that multiple credentials resented must have been issued by to the same master secret.

Zero Knowledge

Zero knowledge is used in the creation of CL signatures. As CL signatures are typically created by two parties, one committing secret information for the other to sign along with some other visible messages. This is certainly the case in anonymous credentials. Zero-knowledge proofs are used to establish trust between the two parties that their secret parts of the signature are done correctly.



The receiver of the signature creates a Zero Knowledge proof of knowledge that they do indeed have knowledge of the secret information they are committing to the signature. In anonymous credentials, they also prove that the master secret they are committing to is the same one used to generate the pseudonym they are known by. They send this to the issuer along with the committed secret information. This is then easily verified by the issuer before they create the signature.

The issuer then creates the signature and builds a proof that the signature was created correctly. The proof and the signature is sent to the receiver who validates the proof against the signature and then can be certain they did indeed receive the correct signature.

CL Signatures in practice

As part of my exploration of CL signatures, I have been dabbling in the code. It helped me to understand the nuances of the signature. The code I looked at was this Java implementation, which while very useful for understanding the code (my java is decent), it was less useful for understanding CL Signatures in terms of anonymous credentials.

Looking at this code in combination with the indy-crypto Rust code which is designed specifically for the anonymous credential use case helped me helped me

to further clarify my understanding. You want to look in the libindy-crypto/cl/issuer.rs && prover.rs files. Here you can find the code relevant for issuing a credential. For a full example of the workflow to issue a credential look in cl/mod.rs. Here you will find tests such as multiple_predicates() running through the issuing of credentials.

The workflow seems to be:

1. Build the credential schema — the list of attribute names that are included in a credential (Although in Sovrin these schemas may already be defined and stored on the ledger)
2. Add in attributes not defined in the schema — ie the master secret attribute
3. The ISSUER generates a public key from the credential definition along with a ZK proof this key is correct for the given credential schema. (To create a correct public key for CL signatures you must define the size of the message blocks — equivalent to the number of attributes for credentials)

they ask for a credential proof of values before issuing a credential.

6. PROVER blinds credential secrets:

- PROVER first verifies proof of credential public keys correctness
- For all hidden attributes in the credential, the PROVER generates a commitment to that value using the credential public key created by the ISSUER
- PROVER generates a proof that these commitments were created correct

7. ISSUER then:

- Verifies the proof of the blinded secrets
- Creates a context for the credential
- Signs the credential
- Generates a proof that the signature is indeed correct for the given credential

8. PROVER then verifies this signature and accepts the credential if correct

I have skipped over some aspects such as credential revocation, but I think this is a great place to start for anyone looking to understand how CL signatures are used in anonymous credential systems.

Originally published at misterwip.uk.

Blockchain Cryptocurrency Ethereum Bitcoin ICO

Follow

Written by Will Abramson

284 Followers · Writer for Good Audience

Enthusiastic software developer excited about the prospects of blockchain technology.

More from Will Abramson and Good Audience

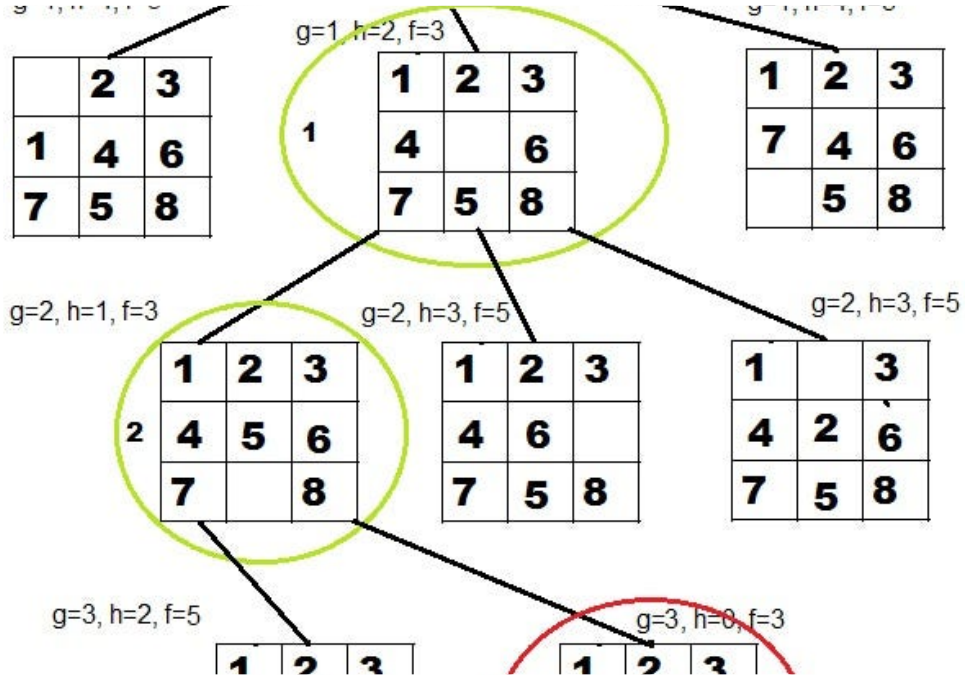
Will Abramson in We've moved to freeCodeCamp.org/news

Lessons learned from deploying my first full-stack web application

I recently achieved one of my long-term goals: deploying my first full-stack web application.

8 min read · Mar 25, 2018

4.1K 18



Ajinkya Sonawane in Good Audience

Solving 8-Puzzle using A* Algorithm.

Solving the sliding puzzle using a basic AI algorithm.

5 min read · Sep 15, 2018

1.1K 15

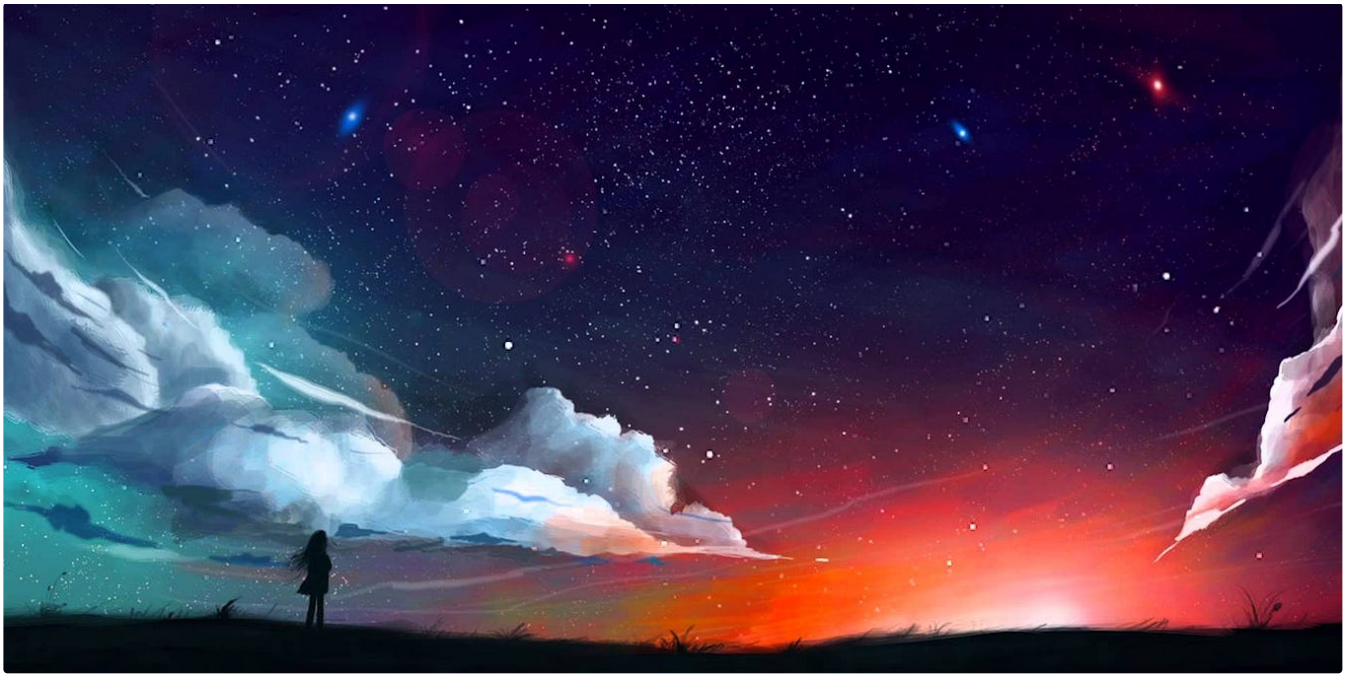
Nils in Good Audience

Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences

An explanatory walkthrough on how to construct a 1D CNN in Keras for time sequences of sensor data.

· 7 min read · Sep 4, 2018

3.7K 33

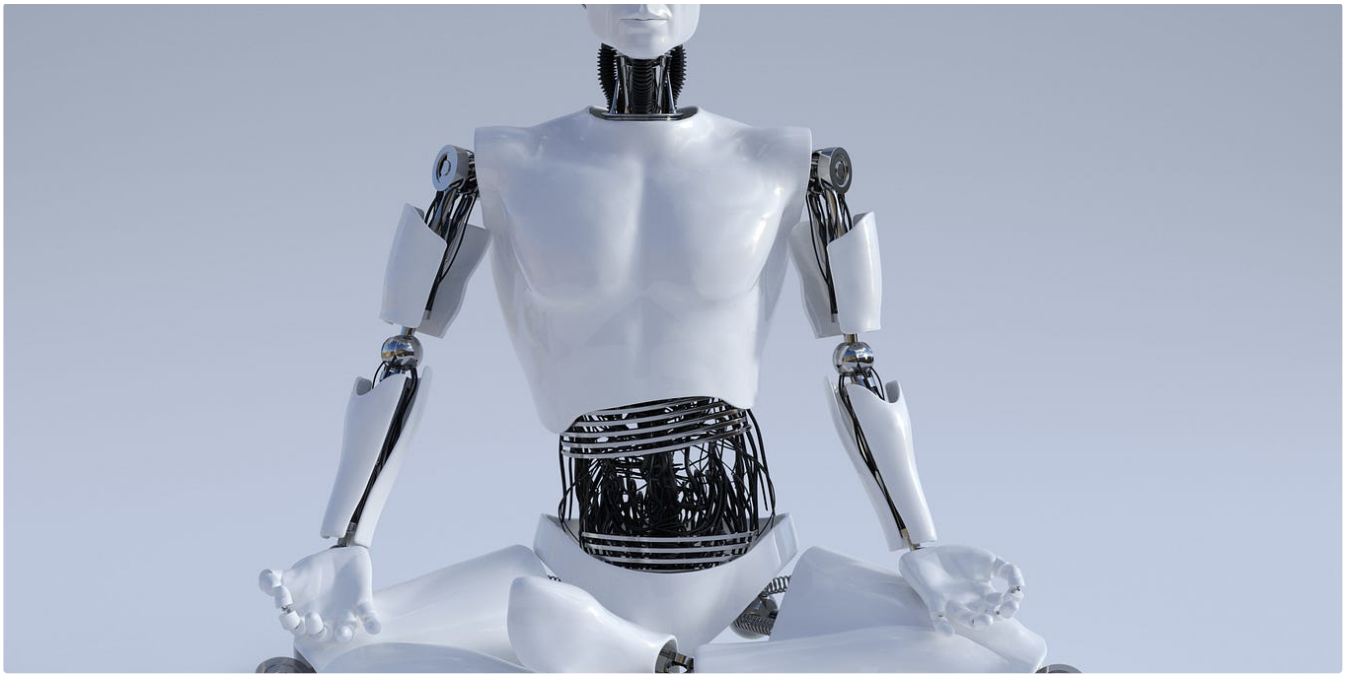


Will Abramson in MyData Journal

Do you exist online?

What aspect of your digital life truly belongs to you?

8 min read · Aug 23, 2018



The PyCoach in Artificial Corner

You’re Using ChatGPT Wrong! Here’s How to Be Ahead of 99% of ChatGPT Users

Master ChatGPT by learning prompt engineering.

· 7 min read · Mar 17

18.3K 327

Unbecoming

10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my morning trail run. I'm wearing my stay-at-home mom...

· 4 min read · Feb 16, 2022

47K 757

Lists

My Kind Of Medium (All-Time Faves)

36 stories · 4 saves

Staff Picks

300 stories · 62 saves

Aleid ter Weel in Better Advice

10 Things To Do In The Evening Instead Of Watching Netflix

Device-free habits to increase your productivity and happiness.

· 5 min read · Feb 15, 2022

18.3K 246

Linda Caroll in The Partnered Pen

I Asked ChatGPT How To Earn $1000 Online. It Was Hilarious.

Peering in the hive mind can be really helpful, but it can also be so stupid it's funny

✨ · 6 min read · Mar 24

9.8K 154

Alexander Nguyen in Level Up Coding

Why I Keep Failing Candidates During Google Interviews...

They don’t meet the bar.

· 4 min read · Apr 13

3.6K 108

Bryan Ye in Better Humans

How To Wake Up at 5 A.M. Every Day

An unconventional and compassionate guide to becoming an early bird

· 15 min read · Oct 3, 2019

86K 638

See more recommendations