2019-04-03 Meeting Notes

Attendees:
- Vipin Bharathan
- Hart Montgomery
- Dave Huseby
- Michael Lodder
- Dan Middleton
- Cam Parra
- Jon Geater
- Manu Driviers
- Brent Zundel
- Maryam Hezaveh
- Lovesh Harchandani

Updates
- Looking for speakers for
  - ICMC '19
  - Consensys Construct
- Ursa update—want to talk about active/incubation status

Discussion
- Language bindings RFC.
  - ML: no global variables in RUST that he knows of; possible in executables but not with libraries; using a map for lookup is not possible—wiped out when call to lib ends
  - DH: Investigate more, we will go forward for now and alter if there are better methods
- Proposal: All maintainers to green check mark on RFCs
  - Dan: idea is good but in practice may be delayed due to 5 more reviewers need.
  - Approved with the caveat that if the delay is too great we will revisit this in a later meeting.
  - (this was not a proposal so much as calling out the defined process see below in comments)
- Encryption API RFC
  - Dan: For all new features I think it's best if we drive them from use cases. In this case it would be helpful to test the proposed API by looking at how the projects are currently using encryption features or how we will be using them in zmix.
  - Dave: I have issues with an interface where we are passing byte arrays and pointers across the FFI boundary, purely on security issues.
  - Mike: Tried to figure out a way to get around it but don't know how to make globally allocated objects work in a Rust library that doesn't require manual ref counting in calling languages. Even languages that have garbage collection.
  - Another idea would be to follow Mozilla's idea of Using protobufs to serialize + deserialize across the FFI boundary
    - only concern is PrKs leakage.
    - could instead pass filenames or key ids across the boundary and have Ursa load the PrK instead of the calling code loading the PrK and passing it across the FFI.
  - We want to use Rust traits to make composable primitives to build up specific combinations of primitives (e.g. AES-GCM).
- National crypto standards discussion
  - Dan: have some concerns of including crypto implementations that people don't think are safe. I have a more general concern about going too wide with wrapping. I would rather we lead with the novel features of zmix rather than add a lot of wrapping that may never be adopted.
  - Dave: the point is not to endorse national standard crypto but to include it so that we have some control over how it is being used.
    - need to make sure that we have strong checks and boundaries to avoid mixing national crypto and other crypto.
    - going to use OpenSSL for national crypto support with an eye towards getting our own regulatory approval eventually
  - Jon: has concerns about using OpenSSL (or any stateful crypto back-end):
    - If we wrap through to OpenSSL for the crypto implementation of SM2, but then to intel built-ins for AES, and some HSM or wallet for ECC, then our overall application will be swimming with loaded states, logins, contexts, sessions...which need to be managed carefully, finalised, destroyed and so on at the right time. If people have concerns about tainting then I think it's much more important to ensure we keep these various memory spaces clean and isolated rather than worrying about particular crypto implementations at the far end of one particular API call.
    - in some situations mixing national crypto and non-national crypto is required, especially going in/out of China. this may require re-design of our checks and boundaries in the build system.
- ZMix RFC
  - one is to be commented on MD two weeks rediscuss
- AnonCreds
  - Section 3.0.2 credential life cycle- Group owner vs. multiple issuers, Holder can send a different secret value - link secret binds them
  - Credential Index- a number, set memberships, merkle trees, cryptographic accumulators.
  - What is the link between this doc(Sovrin specific) and Ursa, needs to be updated to be generalized. (BBS signatures are Sovrin )
  - Puts the algorithms to Bulletproofs paper. only in one algorithm is that made explicit.
- What are we doing for Ursa CI/CD
  - Forming a committee to make a recommendation to TSC by June.
  - K8 cluster looks like the best way forward.