Portable Shared Security Tokens

Source Code

- https://github.com/guardianproject/gnupg-for-android
- https://github.com/guardianproject/gnupg-for-java
- https://github.com/guardianproject/keysync
- https://github.com/guardianproject/notepadbot
- https://github.com/guardianproject/IOCipherServer
- https://github.com/guardianproject/IOCipher
- https://github.com/guardianproject/libsqlfs

Areas of Inquiry

- [[PSST User Stories]]
- [[Encryption and Identity Verification]]
- [[Securely Storing and Syncing]]
- [[Existing PSST Tools]]
- [[Products of PSST Work]]
- related blog posts: https://guardianproject.info/tag/psst/

Workplans

- [[Workplan for Year 2]] "(current)"
- [[Workplan for Year 1]] "(past)"

Background Narrative

While the use of public key-based cryptographic tools (HTTPS/SSL, OpenPGP, OTR chat) has been growing rapidly within human rights and activists organizations, a growing issue looms around the increased diversity of mobile devices that a user might have. The current solution used for managing a set of public and private keys for identifying a circle of trusted people and services is designed around a first-world model ("This computer is mine and no one else's"). As the worldwide computing boom leans decidedly towards mobile devices, new concepts and approaches to managing trust and securing communications need to take into account these new types of users. A portable yet secure solution for establishing one's identity, based upon interoperable standards need to be developed.

This is by far the current most vexing end-user issue we face with our early deployments of Guardian handsets, as well as the most unique aspect of the entire proposal. At this point, we have to create completely new security identities that are separate from any existing identities, keyrings, webs of trust, etc that exist on the desktop. Having identity tokens that can easily sync between computing contexts, with them being secured the entire time is an important problem to work on, that can have some near term end-user benefit. I also believe that just the research into this area, along with published papers, blog posts, and code on the subject, could make a meaningful contribution to the field in general. With our recent breakthroughts in bringing a viable, cross-platform open-source encrypted database, I feel like we've got a big headstart into engineering this, as well. It is one of these features that users aren't asking for specifically, but in the back of their heads they wish they had. We have made some ground with this in Gibberbot (QR code scanning of identity keys, etc), but that is just a first step.

As cryptographic software proves itself to be useful, parts of the infrastructure to maintain it is showing severe weaknesses. The certificate authority model of HTTPS/TLS is being broken in greater ways with each month passing. The PGP "Web of Trust" has proven itself in such communities as Debian, now the barriers for widespread adoption are the ease of use of the software. The problem can be broken down into three parts: the use of keys for signing/encrypting, the signing of other keys to provide identity validation, and keeping the private and public keys in sync on various devices. The first is addressed in a fair amount of software packages like APG, Gibberbot, K-9, Pidgin, Adium, etc. For signing of other people's keys, this is not widely addresses outside of very technical circles. Gibberbot makes it quite easy to verify another person's OTR fingerprint, Seahorse simplifies the PGP signing process quite a bit, but there is no app that handles all signing processes with easily. For the third part, gnupg provides robust public key and signature syncing for PGP keys but that leaves out OTR keys and TLS certs. The PGP Web of Trust also has privacy issues since all of the data is stored on public servers, allowing for easy construction of entire social graphs. In many countries, this would endanger the users of the crypto software.

Since this project requires the syncing and secure storage of files, it might make sense to also expose this syncable, secure file store for general use.