Application Description

Chris Morin <chris.morin2@gmail.com>

Table of Contents

1. Introduction 1
2. Layer specification 1
    2.1. User definition 1
    2.2. Central identity server 2
    2.3. Communication protocol 3
    2.4. Organization applications 4
    2.5. Web interface 4
3. Use cases 5
    3.1. Individual 5
    3.2. Large company 5

1. Introduction

This document will cover the details about each layer of the application stack and how they fit together.

Application stack

- Web interface (optional)
- Organization applications
- Communication protocol
- Central identity server
- User definition

2. Layer specification

2.1. User definition

Each user will have a user number along with a public/private key pair. Their private key will be know only to them while their user number and public key will be held in a central authentication server available to all. We might see the need to give each user a password which is solely used in changing their public key located on the central identity server. This will be useful for cases of users loosing or having their private keys stolen. It might also allow users to change their user number (or create an alias for it) if this is needed.

user number

The user number will be a 64-bit number. This will allow about 1.84e19 unique numbers. The first 4 bits will signify the type of user. At the point of writing this document, the types of users are listed below. This list is preliminary and will likely be changed.

- individual (This might be split up by citizenship status but for now isn’t to avoid possible discrimination)
- federal government organization
- provincial government organization
- municipal government organization
- other government organization
- non-profit NGO
- for profit organization
- other

This user number can be stored in a QR-code for quick and easy tranferal.

public/private key

The public and private key will need to be large enough to ensure security even as computational capacity increases. It might be advantagious to give organizations larger key lengths than indiciduals, but for now we’ll say all key lengths are 3072 bits. The specific encryption algorithm used will be decided at a later time.

> **Note**
>
> It would likely be necessary for users to have to regularly change their public keys.

password

This password will be the user’s way to prove he is the owner of a number in the case that their private key is lost or stolen.

2.2. Central identity server

While the private key will only be known to the user, the public key will be held by the central identity server (CIS). The CIS will contain a list of all existing user numbers and their associated public keys. Any user will be able to query it for the public key associated with a user number. The server might also contain additional user information such as what type of user it is (citizen, government organization, for profit company, …) and their IP address if applicable. The central identity server will have it’s own public-private key pair which is the only thing needed for a user to be able to authenticate the identities of other users.

Users will be able to cache a the public key of other user to speed up communication and be less dependant on the central ID server. This way, if ever the central identification server goes down, organizations will be able to operate as normal save the ability to add new users.

The central identification server will keep a record of each user’s password and this password will give them the ability to modify their public key. The organization applications will regularly query the central server for any public key changes (not added users); this will ensure the old key quickly gets removed from the system.

[NOTE] The Central identity server is really just a user like any other. It has it’s own user number and public/private key. The key difference is that since the CIS is the server that authenticates username/public-key pairs, there is no authority that authenticates the CIS. The public key will need to be gotten from a trusted user who already has the public key. This key will also be widely published on the web. Like other users, it might be necessary to update the CIS’s public key. This might be done on a regular interval (say each week) and the server will authenticate it’s new key with it’s old one.

2.3. Communication protocol

This will be the most complicated part of designedGov. It governs how users will securely share information give modification privledges over their data to other users. The specifics will be described in another document, but I’ll provide an overview here.

Certain special users will have what we will call organization applications. These users, often government organizations or companies, will have their own servers running applications that interact with other users. An example organization with an application would be Canadian Revenue Agency. Their application would keep an account for every tax paying user and users would be able to provide relavant information to them via their application.

Data will be exchanged between users (both individuals and organizations) by opening a TLS session, using their public keys. If the server doesn’t have a cached version of the user’s public key, it will query the CIS to verify it’s authenticity. Once the TLS session is open, they will communicate by passing data in a special format, from here on out called dG format, which will be a subset of the popular JSON format. Each dG messages will be either requests or responses. A request can contain things such as a request for data, an operation to perform on the account, or a granting of permissions. A response relates to a request and can contain data or success/error message.

All data contained in a user account on an application will be translated to a dG object before being sent out. For those unfamiliar with JSON, an object contains a list of key-value pairs and can also contain child objects. This way, each key-value pair can be accessed by traversing a root object. This object will follow a standard which makes it easy for users to see what data is available to them, what they can modify, and what permissions others have over their data. A user can query a server for the information stored on his account and the response will be the root object for that account. This response might point to files other than the root object file when user data is too big to be conviniently placed in a single message. An example of this could be a company’s history of balance sheets in their account with the CRA. Instead of their root object containing all past balance sheets, it contains a list of pointers to objects (one for each financial year). These objects can themselves point to others (one for each month). Some files pointed to in dG files can also be of formats other than dG. These non-dG files will be for when data such as pictures can’t be appropriatly serialized in a dG object. A possible use for this would be storing x-ray images in a patient’s medical record.

A user can make requests to read/modify not only his account but the account of others too. This is the whole point of designedGov: automating data exchange between users. In order to be able to read/modify another user’s account, permission must be granted. By default, all permissions can only be granted by the owner of the account. We will likely need to build a system on top of this that allows for access to data which isn’t granted by the owner of the data. This could be the case in police investigations or emergency access to a patient’s medical record.

2.4. Organization applications

Each organization that wishes to keep information on users and interact with them will host an application. Like any other user, this organization will have it’s own user name and public key, but unlike regular users, they will be able to accept incomming connections from other users through their applications. Each government organization that interacts with the public will have one of these applications. The application will follow the communication protocol when interacting with users. Each application will be responsible for defining their own dG object hierarchy and will make this format public. This way, when a user wants to request a specific key-value pair, they will know the path from the root object. The applications are also resonsible for defining who can read/write what and what permission options are available for each object.

There is nothing stopping non-governmental organizations such as for profit companies from making an organizational application and it is in fact encouraged. This way, all interactions between individuals and organizations will be done via a single secure identity. We won’t need to carry any cards in our wallet anymore (things like your credit and banking accounts will be accessible through your desigedGov ID).

2.5. Web interface

The end user obviously isn’t expected to write dG objects himself, so how will users interact with their various accounts? The answer is that they’ll access their accounts through a web interface. This web application will know the data formats of the most popular government applications and will tailor it’s functionality to suit each of them. It will not only display user account data, but will also give the user the ability to modify it and grant permissions. For convinienve, it will coalesce many small operations into "actions". An example could be selling your car to an individual: the operations of changing car registration with the DMV and declaring the transaction with the CRA would be done in a single step. This action could act on behalf of two users, making the action a contract that each user signs. Coming back to the car example, a transferal of funds from one user’s bank acount to another’s could be part of the car sell.

Note

Some of the higher order actions will be implemented by the organization applications directly instead of the web application coalescing smaller operations. In the case of the car sale, the DMV could themselves report the transaction to the CRA. This would be used when we don’t want to give the user the ability to perform some operations without performing others (if all car sales needed to be reported to the CRA, the DMV would be required to report it themselves).

Although a reference web interface will be implemented as part of designedGov, others will be encouraged to develop their own competing interface for various platforms.

3. Use cases

To get a feel for how all these layers come together, we’ll look at a few fictitional use cases. We’ll get to see two different types of users: an individual and a large company with an organization application.

3.1. Individual

Individuals will be by far most numerous type of users. A user account would be necessary to interact with our society, so every resident and even most visitors would have one.

Today is an important day for Bob, as he’s expecting to hear back from a school he just had a job interview for. In the morning, he hears good news from the school: he got the job! The school principle, who is responsible for administering the hiring process already exchanged user account information with Bob so they have each other’s user number and public keys (Bob doesn’t have the principle’s account details, but the school’s). The principle tells Bob to check his inbox for instructions on how to accept the offer and officially be employed by the school.

Bob decides to take the job and checks his inbox. He sees an attached dG file specifying the details of the employment contract. It has been digitally signed by both the school and the principle. His dG file reader automatically validates their signatures, and his instructions are to digitally sign the contract and send it back. He reads the contract and agrees on it’s conditions and so he signs it (digitally). Inside the contract are a set of actions that will be performed on his accounts, and by signing it, he agrees to them. He sends the contract back to the principle. Within minutes, the principle sends the signed contract off to the CRA’s organization application. The CRA’s application parses the contract and applies all the necessary changes to both the school’s account and Bob’s. At this point, Bob gets a message from the CRA telling him what has been done. Bob doesn’t have to do anything else related to his employment as all the "paper work" has been automated.

3.2. Large company

corporation inc…