Collection process & artifacts.

This note sketches a notion of a scheme for electronic collection of signatures in supporting a referendum innitiative.

The objective of such notion is to help define clear security requirements and assumptions, as well as to facilitate security analysis.

Notation

Parties denoted with **__boldface__**.
Artifacts denoted with *_italics_*.

Parties

1. **__Committee__**: Initiaties an initiative with the objective of collecting enough signautes to support it.
2. **__Chancellerie__**: Verifies the initiative, counts the signatures, attests the end result.
3. **__Collectors__**: Mandated by the Committee to collect signatures
4. **__Voters__**: Give their signatures to support the initiative
5. **__Commune__**: Verifies signatures and ensures each voter contributes at most once

In addition, all parties except **__Voters__** have internal employees, denoted **__XXX.Emp__** where **__XXX__** is the party in question.

The party **__Commune__** conflates Canton/Commune/Electoral committee for simplicity.

Process

In each party **__XXX__** with employees, each **__XXX.Emp__** that intervenes on an internal process of **__XXX__** receives a *_Mandate_* from **__XXX__**. The mandate may be role based, or task-based (a role specific to the initiative/batch of signatures etc.).

1. **__Committee__**: Creates and authenticates *_Initiative_*
    * **__Committee.Emps__** contribute to the Initiative
2. **__Chancellerie__**: Verifies, approves and authenticates the *_Initiative_*
    * Concrete **__Chancellerie.Emps__** perform the verification
3. **__Committee__**: Mandates **__collectors__** with the collections of *_Support signatures_* for the *_Initiative_*
    * Inidividual mandates are issued by **__Committee.Emps__**
4. **__Collectors__**: Collect *_Support signatures_* for the *_Initiative_* from **__Voters__** and transfer them to **__Committee__**
    * Individual *_Support signatures_* are collected by concrete **__Collector.Emps__**
5. **__Committee__**: Send collected *_Support signatures_* to **__Commune__** for verification, in order to obtain *_Signature certificates_*
    * Verification and processing of signature (batches) could be performed by individual **__Commune.Emps__** but possibly also automatized
6. **__Committee__**: Obtains *_Signature certificates_* from the **__Commune__** and keeps track of the total count

7. **Chancellerie**: Obtains all collected *Signature certificates* from **Committee**, verifies them and if the count is sufficient, issues a *Confirmation of the initiative success*.
    * Verification and processing of signature (batches) could be performed by individual **Chancellerie.Emps** but possibly also automatized

Artifacts

The artifacts highlighted in the process may be constructed in many ways, depending on the construction. Additional, auxiliary artifacts may be used if needed, but those above shall be embodied in any distributed construction that aligns with the existing governance structures and procedures.

Security goals

Support count validity

The final count of the signatures considered by the chancelerie shall be no greater than the number of unique, eligible voters who signed in favor. (Or the proportion of the count due to eCOllected signatures, in the multi-modal collecting case.)

For this property, **Commune** and **Chancellerie** are considered honest, while the **Collectors**, **Voters** and potentially also **Committee** may be malicious (the latter can be assumed to have stakes in seeing the initiative pass). We assume none of these parties will want to give up their long-term private keys.

Sketch of a security game

This is modelled with a security experiment, where the **Commune** and **Chancellerie** keys and actions are controlled by a "challenger" and an "adversary" controlls keys and actions of **Collectors**, **Voters** and **Committee**:

1. The challenger initializes keys of **Chancellerie** nad **Committee**
2. The attacker can ask for as many **Communes**, **Collectors**, and **Voters** to be initialized, and gets a unique handle for each. The challenger counts how many unique voters have created _*Support signatures*_ (not how many signatures they created!).
3. The attacker can ask the challenger for any of the **Collectors**, **Voters** and **Committee** to produce their respective artifacts with adversarially-chosen values
4. The attacker can submit arbitrary artifacts to **Commune** and **Chancellerie**, which evaluate them following the protocol.
5. At the end, the challenger lets the attacker win if the accumulated count > the count of unique voters who participated.

Note that (when proven the attacker cannot win with high probability), this property is a sufficient condition for all of the following:

* Collectors cannot submit *Support signatures* that do not originate from eligible voters
* Eligible voters cannot submit duplicit *Support signature* that would all be considered towards the final count

Verifiable *Support signature* inclusion
An eligible **Voter** who cast their *Support signature* shall be able to verify that their signature has contributed towards the final count.

For this, the **Voter** is assumed to have access to
* *Initiative*
* Their own *Support signature*
* *Signature certificates*
    * all?
    * from their commune?
* *Confirmation of the initiative success*

For this property, the adversary may be everyone except the **Voter**.

Initiative-binding
An e-collected signature shall be bound to a concrete initiative, and shall not be reausable.

Employee anonymity
The main artifacts should not leak the identity of concrete employees who processed them

Vote secrecy
The **Chancelerie** shall not know the identities of the voters, nor their votes.

Vote unlinkability

The **Commune** shall not be able to deduce votes of voters and correlate their votes in different initiatives/

Auditability and traceabilty
The construction shall allow a non-repudiable tracing of all artifacts and actions to concrete parties with their collaboration. (In case of an audit and when ordered by court, various parties may reveal some of their private parameters, with which we can confirm their participation/artifacts they produced, artifact aggregation etc).

Multimodel collection
The construction should allow for external sources of (aggregated) *Support signatures*.

Observations

Unlinkability by Commune
The property of unlinkability of votes by the same person between different initiatives may be trivially unachievable for any construction that only collects responses from voters that do support the initiative, if the support of multi-modal collection is required:
* The necessity of verification and double-signature prevention (between paper and eCollecting, for example), likely implies each collected e-*Support Signature* must be linkable to an identity by the Commune
* The collect-only-support paradigm means, that the act of submitting the support leaks the value

* Commune can correlate these signals and create an independent list of supporters for each initiative