Donau Protocol Overview

Johannes Casaburi Pius Loosli Lukas Matyja

April 16, 2024

1 Notation & Definitions

1.1 Notation

- $\langle a, b, ... \rangle$: Pair/tuple

1.2 Definitions

- **Cryptographic Hash Function**

  $$h := H(m)$$

  where m is a message and h the resulting hash.

- **Blinding Function**

  $$\overline{u} := blind(u, b, K_x^{pub})$$

  where u is the value to blind, b the blinding factor to apply and K_x^{pub} the public key of the Donation Unit that will be used for signing.

  The blinding can be done with either the **RSA** blind signature scheme or the Blinded **Clause-Schnorr** signature scheme.

- **Unblinding Function**

  $$\beta := unblind(\overline{\beta}, b, K_x^{pub})$$

  where $\overline{\beta}$ is the value to unblind, b the blinding factor to apply and K_x^{pub} the public key of the Donation Unit that was used for signing.

  The unblinding must be carried out using the **same** signature scheme that has already been used for the blinding.

- **Donation Unit Key generation**

  $$\langle K_x^{pub}, K_x^{priv} \rangle := Keygen^B(\omega)$$

  where ω is a source of entropy. The resulting key pair represents a **Donation Unit**. The result is a public key K_x^{pub} and private key K_x^{priv}. The equivalent used in Taler system is a `Denomination`.

- **Donau Key generation**

  $$\langle D^{pub}, D^{priv} \rangle := Keygen^D(\omega)$$

  where D^{pub} and D^{priv} are the respective public and private Donau keys.

- **Charity Key generation**

  $$\langle C^{pub}, C^{priv} \rangle := Keygen^C(\omega)$$

  where C^{pub} and C^{priv} are the respective public and private Charity keys.

- **Donation Unit (DU)**

$$\langle K_x^{pub}, K_x^{priv} \rangle$$

A Donation Unit consists of a public and private key where x is the associated value (e.g. 2 EUR).

- **Donor Identifier (DI)**

$$i := H(\texttt{TAXID}, S)$$

where S is a random salt with sufficient entropy to prevent guessing attacks to invert the hash function.

- **Unique Donor Identifier (UDI)**

$$u := \langle i, N \rangle$$

where N is a high-entropy nonce to make the resulting hash **unique** per donation.

- **Blinded Unique Donor Identifier (BUDI)**

$$\overline{u} := blind(u, b, K_x^{pub})$$

A **BUDI** is the result of blinding a Unique Donor Identifier u where b is the blinding factor and K_x^{pub} the associated Key. The blinding is done to protect the privacy of the donor.

- **Blinded Unique Donor Identifier Key Pair (BKP)**

$$p := \langle \overline{u}, H(K_x^{pub}) \rangle$$

A **Blinded Unique Donor Identifier Key Pair** is the result of adding the corresponding hash of the **Donation Unit** public key to the **Blinded Unique Donor Identifier** $\overline{u}$ where $H(K_x^{pub})$ is the hash of the **Donation Unit** public key.

- **Signing**
  - **Normal signing (e.g. EdDSA):**

$$\boxed{s := sign(m, k^{priv})} \tag{1}$$

where m is a message and k^{priv} is the private key used to sign the message, for example the Donau private key D^{priv} or the Charity private key C^{priv}.

Applications:

* Signatures over a **Blinded Unique Donor Identifier Key Pair**:

$$\boxed{\vec{\mu}_s := sign(\vec{p}, C^{priv})} \tag{2}$$

where $H(K_x^{pub})$ indicates which **Donation Unit** key should be used by the Donau to sign the resulting **Donation Receipt**. Thus, this hash carries the information about the exact value, the final Donation Receipt should carry.

A charity signs a collection of **Blinded Unique Donor Identifier Key Pairs** before transfering them to the Donau to issue the **Donation Receipts**

* Generation of the **Donation Statement**

– **Blind signing(e.g. RSA/CS):**

$$\boxed{\overline{\beta} := blind\_sign(\overline{u}, K_x^{priv})} \tag{3}$$

where $\overline{u}$ is a blinded value and K_x^{priv} is the private key used to blind sign the message.

Application:

* The Donau blind signs **Blinded Unique Donor Identifiers** received from the Charity with the private key matching the public key in the received **Blinded Unique Donor Identifier Key Pair**

- **Verify Functions**

To verify the signatures m corresponds to the message and s to the signature:

– **normal verify**

$$verify(m, s, P^{pub})$$

where P^{pub} can be the Donau public key D^{pub} or Charity public key C^{pub}.

– **blind verify**

$$verify\_blind(m, s, K_x^{pub})$$

verify a signature that was made blind and made with a Donation Unit private key K_x^{priv}.

- **Donation Receipt**

$$r := \langle u, \beta, H(K_x^{pub}) \rangle$$

where β is the unblinded signature sent to the Donau to get the **Donation Statement**.

- **Donation Statement**

$$\sigma := sign(\langle i, \Sigma\vec{r}, \texttt{Year} \rangle, D^{priv})$$

The **Donation Statement** is the signature over the sum (amount donated) of all the **Donation Receitps** $\Sigma\vec{r}$, that a donor has received from donating throughout the year where i is the **Donor Identifier**.

These signatures attest the amount donated in a particular year by a specific donor.

2 Protocol Detail

2.1 Key generation and initial setup

2.1.1 Initial Donau setup

1. The Donau generates a public key D^{pub} and private key D^{priv} for EdDSA signing.
2. The Donau generates the **Donation Units** consisting of K_x^{pub} and K_x^{priv} where x is the associated value.

2.1.2 Charity setup (Charity side and Donau side)

1. The **Charity** generates a public key (C^{pub} and private key C^{priv}) and fetches the **Donation Unit** public keys from the Donau.
2. The **Charity** transmits C^{pub} and the desired yearly donation limit to the party which maintains the Donau (e.g tax office) using a **secure channel**.
3. The party in charge of Donau administration ensures that the applying charity is authentic and publicly recognized as charity organisation. Furthermore, it ensures that all eventual checks required by law are done. After the verification was successful the Charity public key C^{pub} and requested yearly donation limit are registered.

2.2 During tax period

2.2.1 Donor donates to charity and transmits Unique Donor identifiers (future donation receipts)

1. The donor downloads the **Donation Unit** public keys K_x^{pub} from the Donau for the current year.
2. The donor splits the donation amount into a sum of **Donation Units** offered by the Donau.

   Example: With ***Donation units*** $\{1, 2, 4\}$ *beeing available, and a donation of* 7*, the donation amount is split into the valus* 4*,* 2 *and* 1*.*
3. The donor generates as many **Unique Donor Identifiers** as there are terms in the calculated sum.

   In our example, there are 3 ***Unique Donor Identifiers****: one per* ***Donation Unit****.* [1]

$$i := H(\texttt{TAXID, S})$$

$$u_1 := \langle i, \texttt{N}_1 \rangle$$
$$u_2 := \langle i, \texttt{N}_2 \rangle$$
$$u_3 := \langle i, \texttt{N}_3 \rangle$$

   where S is the salt and N a Nonce.
4. The donor blinds the **Unique Donor Identifiers** using a *different* blinding factor b for every **Unique Donor Identifier**.

[1] If one Donation Unit is present more than once, then there is more than one Unique Donor Identifier required for said Donation Unit. This depends upon the offered Donation Units.

$$\overline{u}_1 := blind(u_1, b_1, K_1^{pub})$$
$$\overline{u}_2 := blind(u_2, b_2, K_2^{pub})$$
$$\overline{u}_3 := blind(u_3, b_3, K_4^{pub})$$

5. So far, the **Unique Donor Identifiers** do not carry information about their value. The *intended effective value is now indicated* by grouping each **Unique Donor Identifier** with the according hash of the **Donation Unit** public key K_x^{pub}.

   Resulting in a **Blinded Unique Donor Identifier Key Pair** or **BKP** for short.

   It is only the *intended effective* value because the value will only be attributed later on with the signature of the Donau.

   *Note: The public key is not in relation with the sequential index of the **BKP**, it only relates to the value of the pair!*

$$\overline{\mu}_1 := \langle \overline{u}_1, h(K_1^{pub}) \rangle$$
$$\overline{\mu}_2 := \langle \overline{u}_2, h(K_2^{pub}) \rangle$$
$$\overline{\mu}_3 := \langle \overline{u}_3, h(K_4^{pub}) \rangle$$

$$\vec{\mu} := \langle \overline{\mu}_1, \overline{\mu}_2, \overline{\mu}_3 \rangle$$

6. The donor sends all **BKP**'s $\vec{\mu}$ as well as the corresponding **payment** to the charity.

2.2.2 Charity sends signed BKP's to Donau

1. The charity verifies that the amount requested (based on the **Donation Unit** public key hash $h(K_x^{pub})$) for signing is **lower or equal** to the effective amount of the donation.
2. The charity signs (using EdDSA) a structure containing all unsigned BKP's coming from the donor.

$$\sigma_c = sign(\vec{\mu}, C^{priv})$$

3. The charity sends this structure $\vec{\mu}$ and the signature σ_c to the Donau.

2.2.3 Donau sends back the blind signed UDI's to charity

1. The Donau:
   (a) verifies the signature σ_c on the structure.

$$verify(\vec{\mu}, \sigma_c, C^{pub})$$

   (b) increments the current amount of donations received per year of the charity. This value is increased by the total amount of the **Blinded Unique Donor Identifier (BUDI)**'s, if the increment does not exceed the annual limit.

(c) blind signs all the **BUDI**'s using the **Donation Unit** private keys K_x^{priv} matching the public keys used in the hash $h(K^{pub})$ which was inturn used in the **BKP**'s.

$$\overline{\beta_1} = blind\_sign(\overline{u}_1, K_1^{priv})$$
$$\overline{\beta_2} = blind\_sign(\overline{u}_2, K_2^{priv})$$
$$\overline{\beta_3} = blind\_sign(\overline{u}_3, K_4^{priv})$$

(d) sends back all created blind signatures $\overline{\beta_1}, \overline{\beta_2}, \overline{\beta_3}$ to the charity.

2. The charity forwards the blind signatures to the donor.
3. The donor verifies the signatures.

$$verify\_blind(u_1, \overline{\beta_1}, K_1^{pub})$$
$$verify\_blind(u_2, \overline{\beta_2}, K_2^{pub})$$
$$verify\_blind(u_3, \overline{\beta_3}, K_4^{pub})$$

4. The donor unblinds the signatures of the **BUDI**'s to get the signatures of the **Unique Donor Identifier (UDI)**'s. This results in a collection of **Donation Receipt (DR)**'s each consisting of the **UDI**, the signature β and the hash of the **Donation Unit** public key $h(K_x^{pub})$.

$$\beta_1 = unblind(\overline{\beta_1}, b_1, K_1^{pub})$$
$$\beta_2 = unblind(\overline{\beta_2}, b_2, K_2^{pub})$$
$$\beta_3 = unblind(\overline{\beta_3}, b_3, K_4^{pub})$$

$$r_1 = \langle UDI_1, \beta_1, h(K_1^{pub}) \rangle$$
$$r_2 = \langle UDI_2, \beta_2, h(K_2^{pub}) \rangle$$
$$r_3 = \langle UDI_3, \beta_3, h(K_4^{pub}) \rangle$$

2.3 After effective tax period: get tax statement for period from Donau

2.3.1 Donor sends the Donation Receipts to the Donau to get the final Donation Statement.

1. The donor sends the collection of all **Donation Receipts** $\{r_1, r_2, r_3\}$ to the Donau. This happens **manually** once per period.

   It is not done continuously to obtain *unlinkability* between the *issuance* of the **Donation Receipts** (which happens upon donation) and their *submission* for the **Donation Statement**.

2. For each **Donation Receipt** the Donau:

   - checks that K_x^{pub} is known.
   - verifies that the signature β is correct using the corresponding public key K_x^{pub}.

- verifies that the **Donor Identifier** is the same as in other **Donation Receipts**.[2]
- verifies that the `nonce` is unique and was not used before by the donor for the corresponding year.

3. The Donau signs over the total `amount` donated by the donor, `year` and **Donor Identifier** and sends the signature and the total amount back to the donor.

   This results in a final signature called the **Donation Statement**.

$$\sigma_s = sign(\langle i, \texttt{amount}_{Total}, \texttt{year}) \rangle, D^{priv})$$

2.3.2 Donor sends the QR Code to a validator (e.g. tax office)

1. The donor generates a QR code which contains the following:

$$\texttt{QR} = \langle \texttt{taxid}, \texttt{salt}, \texttt{year}, \texttt{amount}, \sigma_s \rangle$$

2. The validator scans the QR code and verifies the **Donation Statement** σ_s.

$$verify(\langle i, \texttt{amount}_{Total}, \texttt{year}) \rangle, \sigma_s, D^{pub})$$

[2]With multiple wallets each wallet must simply obtain a separate **Donation Statement**!