Zero-Knowledge Age Restriction for GNU Taler

Özgür Kesim[1], Christian Grothoff[2], Florian Dold[3], and Martin Schanzenbach[4]

[1] Freie Universität Berlin, Germany
[2] Bern University of Applied Sciences, Bern, Switzerland
[3] Taler Systems SA, Erpeldange, Luxembourg
[4] Fraunhofer AISEC, München, Germany

Abstract. We propose a design for a privacy-friendly method of age restriction in e-commerce that is aligned with the principle of subsidiarity. The design is presented as an extension of a privacy-friendly payment protocol with a zero-knowledge scheme that cryprographically augments coins for this purpose. Our scheme enables buyers to prove to be of sufficient age for a particular transaction without disclosing it. Our modification preserves the privacy and security properties of the payment system such as the anonymity of minors as buyers as well as unlinkability of transactions. We show how our scheme can be instantiated with ECDSA as well with a variant of EdDSA, respectively, and how it can be integrated with the GNU Taler payment system. We provide formal proofs and implementation of our proposal. Key performance measurements for various CPU architectures and implementations are presented.

1 Introduction

Youth protection regulation requires retailers to assist caretakers in their efforts to keep minors safe online [Pou11]. For example, the *Council of Europe Recommendation Rec (2001)8* says that "11. Member states should encourage the use of conditional access tools by content and service providers in relation to content harmful to minors, such as age-verification systems, ...".

Age verification in e-commerce today is mostly implemented by identity verification where the customer has to provide official identity documents. This approach is expensive for retailers, because they have to handle confidential information very carefully, and invasive for customers, because ID cards reveal more than the age. Another, privacy-friendly approach is the use of attribute-based credentials [Kon+14; Sch+19] where authorities issue consumers with a certificate that enables them to produce a zero-knowledge proof [GMR89] showing that they are of sufficient age. A third approach ties age restriction to the ability to pay, for example via specialized credit cards for children that limit them to buy at certain "safe" stores [Fea21], but is also not privacy-friendly.

What all approaches so far have in common is that they violate the *principle of subsidiarity* [Bos10; Pav21], by which functions of government—such as granting and restricting rights—should be performed at the lowest level of authority possible, as long as they can be performed adequately. In case of age restriction, the lowest level of authority is that of legal responsibility for an under-age person: the parents, guardians and caretakers—not government or financial institutions.

Our contribution is the design of an age restriction scheme that combines the following goals:

1. It ties age restriction to the ability to pay, not to IDs,
2. maintains anonymity of buyers,
3. maintains unlinkability of transactions and
4. is aligned with the principle of subsidiarity.

Specifically, we define a zero-knowledge age-restriction scheme as an extension to GNU Taler, a privacy-preserving payment system where consumers can make unlinkable, untraceable payments using digital coins that were blindly signed by the payment service provider [Cha89; Dol19]. GNU Taler as the underlying payment system is in full concordance with our goals.

Next, we will give the formal definition of the age restriction scheme and the security properties of our protocol (Section 2), a specific design and instantiation (Section 3), and security proof (Section 4). We then provide a brief primer on GNU Taler (Section 5), followed by a description on how to integrate the construction into it (Section 6) and assess the impact on performance (Section 7). Finally, we discuss how the assumption on checking accounts being always under the control of adults could be lifted by a small variation of the protocol (Section 8).

We thank Matthias Wählisch for constructive feedback on an earlier draft.

2 Age Restriction

Our design for an age restriction scheme is based on the following assumptions and scenarios:

1. Checking accounts are always under the control of an eligible adult. When such an adult acts as the legal guardian for a minor and provides the minor with digital coins, our system allows them to add age-restriction to the nascent coins as they are being placed into the minor's digital wallet. Subsidiarity is therefore preserved.
2. The minor can then freely and anonymously spent the coins. However, if a merchant requires a proof that the buyer is of a certain age, the minor can only generate zero-knowledge proofs up to the age limit set by their legal guardian. We note that the proofs are tied to each specific coin, allowing the guardian to grant exceptions for certain amounts.
3. The protocol design must also maintain GNU Taler's critical capability to render change (or give refunds) in an unlinkable way, see section 5. When minors receive fresh coins from change or refunds, the age restrictions should carry over to the fresh coins created by those business processes. The protocol must preserve unlinkability, so that it is impossible for merchants or the payment service provider to link the different transactions, even if a minor makes subsequent purchases from coins that were rendered as age-restricted change.

Our design for an age restriction protocol involves the following computations by several parties. First, the *legal guardian* initially withdraws the digital coins and **commits** to an age restriction. Next, the *minor* wants to make a purchase and must **attest** their adequate age. The *merchant* will then need to **verify** the age proof. If the *minor* is to receive change, they must **derive** equivalent age restrictions for the fresh coins, and finally the *payment service provider* must **compare** the age restrictions to ensure that the minor preserved them correctly.

We will begin by giving the signatures for these five functions, then formally state the security requirements and then follow this up with a possible instantiation and a proof that the instantiation satisfies the security requirements.

2.1 Signatures

Let λ be the general security parameter (written 1^λ in unary representation) and $\Omega = \{0,1\}^\lambda$. Let M be the minimum age of an unrestricted adult in years (with M small, typically $\mathsf{M} \in \{18, 21\}$). Then we define an age restriction scheme as the five functions

$$(\mathsf{Commit}, \mathsf{Attest}, \mathsf{Verify}, \mathsf{Derive}, \mathsf{Compare})$$

along with appropriate domains $(\mathbb{P}, \mathbb{O}, \mathbb{T}, \mathbb{B})$, with the following signatures:

$$\mathsf{Commit}: \quad \mathbb{N}_\mathsf{M} \times \Omega \to \mathbb{O} \times \mathbb{P}, \qquad (\mathsf{a}, \omega) \mapsto (\mathsf{Q}_{(\mathsf{a},\omega)}, \mathsf{P}_{(\mathsf{a},\omega)}) \tag{1}$$

$$\mathsf{Attest}: \quad \mathbb{N}_\mathsf{M} \times \mathbb{O} \times \mathbb{P} \to \mathbb{T} \cup \{\perp\}, \qquad (\mathsf{m}, \mathsf{Q}, \mathsf{P}) \mapsto \mathsf{T}_{(\mathsf{m},\mathsf{Q},\mathsf{P})} \tag{2}$$

$$\mathsf{Verify}: \quad \mathbb{N}_\mathsf{M} \times \mathbb{O} \times \mathbb{T} \to \mathbb{Z}_2, \qquad (\mathsf{m}, \mathsf{Q}, \mathsf{T}) \mapsto b \tag{3}$$

$$\mathsf{Derive}: \quad \mathbb{O} \times \mathbb{P} \times \Omega \to \mathbb{O} \times \mathbb{P} \times \mathbb{B}, \qquad (\mathsf{Q}, \mathsf{P}, \omega) \mapsto (\mathsf{Q}'_\omega, \mathsf{P}'_\omega, \beta_\omega) \tag{4}$$

$$\mathsf{Compare}: \quad \mathbb{O} \times \mathbb{O} \times \mathbb{B} \to \mathbb{Z}_2, \qquad (\mathsf{Q}, \mathsf{Q}', \beta) \mapsto b \tag{5}$$

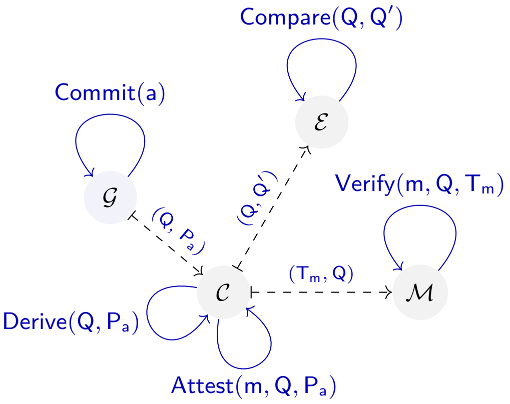
where $\mathbb{P}, \mathbb{O}, \mathbb{T}, \mathbb{B}$ are sufficiently large sets, not prone to exhaustive search. Helpful mnemonics for the sets and symbols are: $\mathbb{O}$ = *c*$\mathbb{O}$*mmitments*, Q = *Q-mitment* (commitment), $\mathbb{P}$ = $\mathbb{P}$*roofs*, P = P*roof*, $\mathbb{T}$ = *a*$\mathbb{T}$*testations*, T = *a*T*testation*, $\mathbb{B}$ = $\mathbb{B}$*lindings*, β = β*linding*. No assumptions about relationships of these sets are being made a-priori, except for $\perp \notin \mathbb{T}$.

Figure 1 shows the function of this scheme as they are called by which various participants and the transferred data between them.

2.2 Achieving unlinkability

In a naïve use of Derive and $\mathsf{Compare}$, children would iteratively call Derive and an exchange $\mathcal{E}$ would call $\mathsf{Compare}$. A child $\mathcal{C}$ would thereby create a chain $\mathsf{Q}_0, \mathsf{Q}_1, \ldots$ of equivalent age commitments and $\mathcal{E}$ would call $\mathsf{Compare}(\mathsf{Q}_i, \mathsf{Q}_{i+1}, .)$ successively to check their validity. However, this would allow $\mathcal{E}$ to recognize the whole sequence $\{\mathsf{Q}_0, \mathsf{Q}_1, \ldots\}$ as being linked to the same $\mathcal{C}$ and therefore violate any requirement for unlinkability of age restrictions and violate indistinguishability of children.

Fig. 1: *Age restriction flow* – The functions (1)-(5) are called by various participants of a payment system: guardian $\mathcal{G}$, child $\mathcal{C}$, exchange $\mathcal{E}$ and merchant $\mathcal{M}$. Compared to the participants in GNU Taler (section 5), the guardian is introduced as a new entity and is responsible for the Commit. The diagram also shows the data being transferred between the participants. Note that the seeds and blindings are omitted for better readability.


$\mathsf{Compare}(\mathsf{Q}, \mathsf{Q}')$
$\mathsf{Commit}(\mathsf{a})$
$\mathcal{E}$
$\mathsf{Verify}(\mathsf{m}, \mathsf{Q}, \mathsf{T_m})$
$\mathcal{G}$
$(\mathsf{Q}, \mathsf{Q}')$
$(\mathsf{Q}, \mathsf{P_a})$
$(\mathsf{T_m}, \mathsf{Q})$
$\mathcal{C}$
$\mathcal{M}$
$\mathsf{Derive}(\mathsf{Q}, \mathsf{P_a})$
$\mathsf{Attest}(\mathsf{m}, \mathsf{Q}, \mathsf{P_a})$

In order to achieve unlinkability and indistinguishably, we extend the functions (1)-(5) and propose a zero-knowledge, cut-and-choose protocol, based on Derive and $\mathsf{Compare}$, in which $\mathcal{C}$ and $\mathcal{E}$ participate in an interactive proof (with a certain success-probability) for honest derivation of a new age commitment from an existing one, without revealing the new age commitment. This protocol follows the design of the $\mathsf{refresh}$ protocol in GNU Taler [Dol19, §4.7.4].

Given $\kappa \in \mathbb{N}$, we define the protocol $\mathsf{DeriveCompare}_\kappa : \mathbb{O} \times \mathbb{P} \times \Omega \to \{0, 1\}$ between the two parties $\mathcal{C}$ (child) and $\mathcal{E}$ (exchange), with H a hash function and uniformly random sampling $\xleftarrow{\$}$, as follows:

$$\mathsf{DeriveCompare}_\kappa(\mathsf{Q}, \mathsf{P}, \omega) := \tag{6}$$

```
$\mathcal{C}$:  1. for all $i \in \{1, \dots, \kappa\} : (\mathsf{Q}_i, \mathsf{P}_i, \beta_i) \leftarrow \mathsf{Derive}(\mathsf{Q}, \mathsf{P}, \omega + i)$
     2. $h \leftarrow \mathsf{H}\big(\mathsf{H}(\mathsf{Q}_1, \beta_1) \parallel \cdots \parallel \mathsf{H}(\mathsf{Q}_\kappa, \beta_\kappa)\big)$
     3. sent $(\mathsf{Q}, h)$ to $\mathcal{E}$
$\mathcal{E}$:  4. save $(\mathsf{Q}, h)$
     5. $\gamma \xleftarrow{\$} \{1, \dots, \kappa\}$
     6. sent $\gamma$ to $\mathcal{C}$
$\mathcal{C}$:  7. $h'_\gamma \leftarrow \mathsf{H}(\mathsf{Q}_\gamma, \beta_\gamma)$
     8. $\mathbf{E}_\gamma \leftarrow \big[(\mathsf{Q}_1, \beta_1), \dots, (\mathsf{Q}_{\gamma-1}, \beta_{\gamma-1}), \perp, (\mathsf{Q}_{\gamma+1}, \beta_{\gamma+1}), \dots, (\mathsf{Q}_\kappa, \beta_\kappa)\big]$
     9. sent $(\mathbf{E}_\gamma, h'_\gamma)$ to $\mathcal{E}$
$\mathcal{E}$: 10. for all $i \in \{1, \dots, \kappa\} \setminus \{\gamma\} : h_i \leftarrow \mathsf{H}(\mathbf{E}_\gamma[i])$
    11. if $h \overset{?}{\neq} \mathsf{H}(h_1 \| \dots \| h_{\gamma-1} \| h'_\gamma \| h_{\gamma+1} \| \dots \| h_{\kappa-1})$ return 0
    12. for all $i \in \{1, \dots, \kappa\} \setminus \{\gamma\}$: if $0 \overset{?}{=} \mathsf{Compare}(\mathsf{Q}, \mathsf{Q}_i, \beta_i)$ return $0$
    13. return 1
```

With this protocol, $\mathcal{E}$ learns nothing about Q_γ (except for the blinded hash $H(\mathsf{Q}_\gamma, \beta_\gamma)$) and trusts it to be of the same maximum age as the original commitment with certainty $\frac{\kappa-1}{\kappa}$. Correspondingly, $\mathcal{C}$ has a chance of $\frac{1}{\kappa}$ to cheat successfully, by using *one* age commitment generated via Commit with a higher age limit, instead of calling Derive on the old age commitment.

2.3 Requirements imposed on the functions (1)-(5)

For the cryptosystem to have the desired intuitive effect of providing age restrictions on purchases for minors, the five functions (1)-(5) must have the properties detailed in this section.

Requirement 1 (Existence of lower bound proofs)

$$\underset{\substack{\mathsf{a}\in\mathbb{N}_\mathsf{M}\\ \omega\in\Omega}}{\forall} : \mathsf{Commit}(\mathsf{a}, \omega) =: (\mathsf{Q}, \mathsf{P}) \implies \mathsf{Attest}(\mathsf{m}, \mathsf{Q}, \mathsf{P}) = \begin{cases} \mathsf{T} \in \mathbb{T}, & \textit{if } \mathsf{m} \le \mathsf{a} \\ \bot & \textit{otherwise} \end{cases}$$

Requirement 2 (Efficacy of lower bounds proofs)

$$\mathsf{Verify}(\mathsf{m}, \mathsf{Q}, \mathsf{T}) = \begin{cases} 1, \textit{if } \underset{\mathsf{P}\in\mathbb{P}}{\exists} : \mathsf{Attest}(\mathsf{m}, \mathsf{Q}, \mathsf{P}) = \mathsf{T} \\ 0 \ \textit{otherwise} \end{cases}$$

Requirements 1 and 2 imply

Corollary 1 (Efficacy of commitments and proofs). *Let* $(\mathsf{Q}, \mathsf{P}) \leftarrow \mathsf{Commit}(\mathsf{a}, \omega)$ *with* $\mathsf{a} \in \mathbb{N}_\mathsf{M}$ *and* $\omega \in \Omega$*. If Requirements 1 and 2 hold, then also*

$$\forall_{n\le\mathsf{a}} : \mathsf{Verify}\big(n, \mathsf{Q}, \mathsf{Attest}(n, \mathsf{Q}, \mathsf{P})\big) = 1.$$

Furthermore, the functions must be related by the following requirements:

Requirement 3 (Derivability of commitment and proofs) *Let* $\mathsf{a} \in \mathbb{N}_\mathsf{M}$*,* $\omega_0, \omega_1 \in \Omega$*,* $(\mathsf{Q}_0, \mathsf{P}_0) \leftarrow \mathsf{Commit}(\mathsf{a}, \omega_0)$ *and* $(\mathsf{Q}_1, \mathsf{P}_1, \beta) \leftarrow \mathsf{Derive}(\mathsf{Q}_0, \mathsf{P}_0, \omega_1)$*. Then*

$$\mathsf{Compare}(\mathsf{Q}_0, \mathsf{Q}_1, \beta) = 1 \tag{7}$$

and for all $n \le \mathsf{a}$*:*

$$\mathsf{Verify}(n, \mathsf{Q}_1, \mathsf{Attest}(n, \mathsf{Q}_1, \mathsf{P}_1)) = \mathsf{Verify}(n, \mathsf{Q}_0, \mathsf{Attest}(n, \mathsf{Q}_0, \mathsf{P}_0))$$

We also do require the converse of (7), that is

Requirement 4 (Surjectivity of Derivation)

$$\underset{\substack{\mathsf{Q},\mathsf{Q}'\in\mathbb{O}\\ \beta\in\mathbb{B}}}{\forall} : \Big(\mathsf{Compare}(\mathsf{Q}, \mathsf{Q}', \beta) = 1 \Rightarrow \underset{\substack{\mathsf{P},\mathsf{P}'\in\mathbb{P}\\ \omega\in\Omega}}{\exists} : (\mathsf{Q}', \mathsf{P}', \beta) = \mathsf{Derive}(\mathsf{Q}, \mathsf{P}, \omega)\Big)$$

We will now define our security and privacy requirements in the form of security games. In the following, λ refers to the general security parameter in unary representation and $\Omega = \{0,1\}^\lambda$. We write $x \overset{\$}{\leftarrow} X$ for a sample x taken randomly from a uniform distribution over X and $\mathbb{N}_\mathsf{M} = \{1, \ldots, \mathsf{M}\}$. $\mathfrak{A}(X \to Y)$ is the set of all probabilistic polynomial-time algorithms from X to Y and $\epsilon(x)$ represents a negligible function, i.e. $\epsilon(x) = 1/O(e^x)$.

First, we will formalize that the age-restriction protocol must not disclose unnecessary information about the age of the minor. Specifically, neither a commitment $\mathsf{Q} \in \mathbb{O}$ nor a related attestation $\mathsf{T} \in \mathbb{T}$ should disclose the age a that went into the first Commit, beyond what is fundamentally disclosed by the age being sufficient to satisfy the age check. We formalize this via the following games and requirements.

Game 1 (Age disclosure by commitment or attestation)

Let $n \in \mathbb{N}^+$, $\mathsf{m} \in \mathbb{N}_\mathsf{M}$ *and* $\mathcal{A} : \mathbb{N}_\mathsf{M} \times \mathbb{T}^n \times \mathbb{O}^n \times \mathbb{B}^{n-1} \to \mathbb{N}_\mathsf{M}$ *(with* $\mathbb{B}^0 := \{\bot\}$*). The game* $G_\mathcal{A}^\mathsf{AgeCA}(\lambda, \mathsf{m}, n)$ *is defined as:*

1. $(\mathsf{a}, \omega_1, \ldots, \omega_n) \overset{\$}{\leftarrow} \{\mathsf{m}, \ldots, \mathsf{M}\} \times \Omega^n$
2. $(\mathsf{Q}_1, \mathsf{P}_1) \leftarrow \mathsf{Commit}(\mathsf{a}, \omega_1)$
3. *If* $n > 1$, *apply for* $i \in \{1, \ldots, n-1\}$*:* $(\mathsf{Q}_{i+1}, \mathsf{P}_{i+1}, \beta_{i+1}) \leftarrow \mathsf{Derive}(\mathsf{Q}_i, \mathsf{P}_i, \omega_{i+1})$
4. *For* $i \in \{1, \ldots, n\}$ *apply:* $\mathsf{T}_i \leftarrow \mathsf{Attest}(\mathsf{m}, \mathsf{Q}_i, \mathsf{P}_i)$
5. *If* $n > 1$, *set* $\mathsf{b} \leftarrow \mathcal{A}(\mathsf{m}, \mathsf{T}_1, \ldots, \mathsf{T}_n, \mathsf{Q}_1, \ldots, \mathsf{Q}_n,\ \beta_2, \ldots, \beta_n)$ *else* $\mathsf{b} \leftarrow \mathcal{A}(\mathsf{m}, \mathsf{T}_1, \mathsf{Q}_1, \bot)$
6. *Return* 1 *if* $\mathsf{b} = \mathsf{a}$ *and otherwise* 0.

Requirement 5 (Nondisclosure of age) *A set of functions with signatures (1)-(5) is said to satisfy* nondisclosure of age*, if for all* $n \in \mathbb{N}^+$*:*

$$\underset{\mathcal{A} \in \mathfrak{A}(\mathbb{N}_\mathsf{M} \times \mathbb{T}^n \times \mathbb{O}^n \times \mathbb{B}^{n-1} \to \mathbb{N}_\mathsf{M})}{\forall} : \Pr\Big[G_\mathcal{A}^\mathsf{AgeCA}(\lambda, \mathsf{m}, n) = 1\Big] \leq \frac{1}{\mathsf{M} - \mathsf{m} + 1} + \epsilon(\lambda) \quad (8)$$

For effective age-restriction, we clearly also need the property that after a call to $\mathsf{Commit}(\mathsf{a}, \omega)$ with an age a, it should not be possible to forge an attest for a higher age from the commitment. This is described by the following game and requirement.

Game 2 (Forging an attest) *Let* $n \in \mathbb{N}^+$, $\mathcal{A} : \mathbb{N}_\mathsf{M} \times \mathbb{O} \times \mathbb{P} \times \Omega^{n-1} \to \mathbb{N}_\mathsf{M} \times \mathbb{T}$ *(with* $\Omega^0 := \{\bot\}$*). The game* $G_\mathcal{A}^\mathsf{FA}(\lambda, n)$ *is defined as:*

1. $(\mathsf{a}, \omega_1, \ldots, \omega_n) \overset{\$}{\leftarrow} \mathbb{N}_{\mathsf{M}-1} \times \Omega^n$
2. $(\mathsf{Q}_1, \mathsf{P}_1) \leftarrow \mathsf{Commit}(\mathsf{a}, \omega_1)$
3. *If* $n > 1$, *for* $i \in \{1, \ldots, n-1\}$*:* $(\mathsf{Q}_{i+1}, \mathsf{P}_{i+1}, \_) \leftarrow \mathsf{Derive}(\mathsf{Q}_i, \mathsf{P}_i, \omega_{i+1})$
4. *If* $n > 1$ $(\mathsf{m}, \mathsf{T}) \leftarrow \mathcal{A}(\mathsf{a}, \mathsf{Q}_1, \mathsf{P}_1, \omega_2, \ldots, \omega_n)$, *else* $(\mathsf{m}, \mathsf{T}) \leftarrow \mathcal{A}(\mathsf{a}, \mathsf{Q}_1, \mathsf{P}_1, \bot)$
5. *Return 0 if* $\mathsf{m} \leq \mathsf{a}$
6. *Return* $\mathsf{Verify}(\mathsf{m}, \mathsf{Q}_n, \mathsf{T})$

Requirement 6 (Unforgeability of minimum age) *A set of functions with signatures (1)-(5) is said to satisfy* unforgeability of minimum age*, if for all $n \in \mathbb{N}^+$ the following holds:*

$$\underset{\mathcal{A}\in\mathfrak{A}(\mathbb{N}_\mathsf{M}\times\mathbb{O}\times\mathbb{P}\times\Omega^{n-1}\to\mathbb{N}_\mathsf{M}\times\mathbb{T})}{\forall} : \Pr\Big[G^{\mathsf{FA}}_{\mathcal{A}}(\lambda, n) = 1\Big] \leq \epsilon(\lambda). \tag{9}$$

Finally, we define a game to challenge the unlinkability of commitments and attestations in which the adversary is considered to be a collaboration of exchange and merchant. Basically, any initial age commitment Q_0 and all its derived successors Q_i – together with all the attestations T_i they were used for – must be indistinguishable from any other such chain.

As argued before in section 2.2, we assume that the cut-and-choose protocol $\mathsf{DeriveCompare}_\kappa$ is being performed between the client $\mathcal{C}$ and exchange $\mathcal{E}$ to guarantee the unlinkability of age commitments in the exchange. This explains the complicated definition of the game, in which we model the execution of $\mathsf{DeriveCompare}_\kappa$ via the data generated from the various calls to Derive and $\mathsf{Compare}$, which are partially made accesible to the adversary, as well as data from Attest.

Game 3 (Distinguishing derived commitments and attestations) *Let $n, \kappa \in \mathbb{N}^+$, $\mathbb{K} := \mathbb{O} \times \mathbb{B}$, $\mathbb{H} := \mathbb{O} \times \mathbb{T} \times \mathbb{K}^{\kappa-1}$, $\mathcal{A}_0 : \mathbb{N}_\mathsf{M}^2 \to \mathbb{N}_\mathsf{M}^{n+1}$ and $\mathcal{A}_1 : \mathbb{N}_\mathsf{M}^{n+1} \times \mathbb{H}^{2n+1} \to \{0,1\}$. The game $G^{\mathsf{DCA}}_{\mathcal{A}_0,\mathcal{A}_1}(\lambda, \kappa, n)$ is then defined as follows:*[1]

1. $\left(\mathsf{a}^0, \omega^0, \mathsf{a}^1, \omega^1\right) \xleftarrow{\$} (\mathbb{N}_\mathsf{M} \times \Omega)^2$
2. $(\mathsf{Q}^0_1, \mathsf{P}^0_1) \leftarrow \mathsf{Commit}(\mathsf{a}^0, \omega^0)$, $(\mathsf{Q}^1_1, \mathsf{P}^1_1) \leftarrow \mathsf{Commit}(\mathsf{a}^1, \omega^1)$
3. *Recursively for $i \in \{1, \dots, n\}$:*

   $(\zeta_i, \eta_i) \xleftarrow{\$} \Omega \times \Omega$

   $\left(\mathsf{Q}^0_{i+1}, \mathsf{P}^0_{i+1}, \_\right) \leftarrow \mathsf{Derive}(\mathsf{Q}^0_i, \mathsf{P}^0_i, \zeta_i)$

   $\left(\mathsf{Q}^1_{i+1}, \mathsf{P}^1_{i+1}, \_\right) \leftarrow \mathsf{Derive}(\mathsf{Q}^1_i, \mathsf{P}^1_i, \eta_i)$

In step 3 we model the part of the cut&choose protocol where one pair of commitment and blinding is ***not revealed*** *to the adversary. The sequences of pairs ($\mathsf{Q}^{0/1}_j$, $\mathsf{P}^{0/1}_j$) in this step are later used for attestation, but the blindings from the calls to* $\mathsf{Derive}()$ *are ignored and not accessible to the adversary.*

4. *For $i \in \{1, \dots, n+1\}$:*

   For $k \in \{1, \dots, \kappa-1\}$:

   $(\xi_k, \chi_k) \xleftarrow{\$} \Omega \times \Omega$

   $\left(A^k_i, \_, \alpha^k_i\right) \leftarrow \mathsf{Derive}(\mathsf{Q}^0_i, \mathsf{P}^0_i, \xi_k)$

   $\left(B^k_i, \_, \beta^k_i\right) \leftarrow \mathsf{Derive}(\mathsf{Q}^1_i, \mathsf{P}^1_i, \chi_k)$

   $\vec{R}^0_i := \left((A^1_i, \alpha^1_i), \dots, (A^\kappa_i, \alpha^\kappa_i)\right)$

   $\vec{R}^1_i := \left((B^1_i, \beta^1_i), \dots, (B^\kappa_i, \beta^\kappa_i)\right)$

In step 4, all ***revealed*** *commitments and blindings during the cut&choose protocol are modelled. The adversary will see $2n(\kappa-1)$ derived pairs (A^k_i, α^k_i) and (B^k_i, β^k_i) of commitments and blindings. The derived proofs are ignored as they are not used for attestation, and not accesible to the adversary.*

[1] Upper indices on variables are not exponents.

5. $(b, i_0, i_1) \xleftarrow{\$} \{0,1\} \times \{1, \dots, n+1\}^2$ — *i_0 and i_1 are random indices that are dropped from each history, respectivelly, in step 9.*
6. $(\mathsf{m}_1, \dots, \mathsf{m}_{n+1}) \leftarrow \mathcal{A}_0(\mathsf{a}^0, \mathsf{a}^1, i_0, i_1)$ — *The adversary chooses minimal ages.*
7. *Return* 0 *if* $\exists_i : \mathsf{m}_i > \mathsf{min}(\mathsf{a}^0, \mathsf{a}^1)$ *or* $\mathsf{m}_{i_0} \neq \mathsf{m}_{i_1}$ — *The minimum ages must not distinguish between* a^0 *and* a^1 *and must be the same at the dropped indices.*
8. $\forall_{i \in \{1,\dots,n+1\}}$*:* $\mathsf{T}^0_i \leftarrow \mathsf{Attest}(\mathsf{m}_i, \mathsf{Q}^0_i, \mathsf{P}^0_i)$*,* $\mathsf{T}^1_i \leftarrow \mathsf{Attest}(\mathsf{m}_i, \mathsf{Q}^1_i, \mathsf{P}^1_i)$
9. $s \leftarrow \mathcal{A}_1\Big(\mathsf{m}_1, \dots, \mathsf{m}_{n+1}, i_0, i_1, (\mathsf{Q}^b_{i_b}, \mathsf{T}^b_{i_b}, \vec{R}^b_{i_b}),$
   $(\mathsf{Q}^0_1, \mathsf{T}^0_1, \vec{R}^0_1), \dots, \cancel{(\mathsf{Q}^0_{i_0}, \mathsf{T}^0_{i_0}, \vec{R}^0_{i_0})}, \;\dots\;, (\mathsf{Q}^0_{n+1}, \mathsf{T}^0_{n+1}, \vec{R}^0_{n+1}),$
   $(\mathsf{Q}^1_1, \mathsf{T}^1_1, \vec{R}^1_1), \;\dots\;, \cancel{(\mathsf{Q}^1_{i_1}, \mathsf{T}^1_{i_1}, \vec{R}^1_{i_1})}, \dots, (\mathsf{Q}^1_{n+1}, \mathsf{T}^1_{n+1}, \vec{R}^1_{n+1})\Big)$
10. *Return* 1*, if* $s = b$*, and otherwise* 0*.*

Requirement 7 (Unlinkability of commitments and attestations) *A set of functions with signatures (1)-(5) is said to satisfy* (unbounded) unlinkability of commitments and attestations*, if for all* $n, \kappa \in \mathbb{N}^+$ *the following holds:*

$$\mathop{\forall}_{\substack{\mathcal{A}_0 \in \mathfrak{A}\left(\mathbb{N}^2_{\mathsf{M}} \times \{1,\dots,n+1\}^2 \to \mathbb{N}^{n+1}_{\mathsf{M}}\right) \\ \mathcal{A}_1 \in \mathfrak{A}\left(\mathbb{N}^{n+1}_{\mathsf{M}} \times \{1,\dots,n+1\}^2 \times \mathbb{H}^{2n+1} \to \{0,1\}\right)}} : \Pr\Big[G^{\mathsf{DCA}}_{\mathcal{A}_0,\mathcal{A}_1}(\lambda, \kappa, n) = 1\Big] = \frac{1}{2} - \epsilon(\lambda) \qquad (10)$$

3 Instantiation with ECDSA

We can now define a general instantiation of (1)-(5) based on ECDSA[2] – general in the sense that the elliptic curve, hash function and generator are variables in the scheme. For the definitions and notations regarding elliptic curves and ECDSA we follow [JMV01].

Let $\lambda \in \mathbb{N}$ be the security parameter, $\mathsf{M} \in \mathbb{N}_+$ the number of age groups to be handled in the system, $E = (\mathbb{E}(p, a, b), G, g)$ be an elliptic curve over the field $\mathbb{F}_p$ with generator G of prime order g with $\log_2 g \geq \lambda$, $\breve{g} := \lfloor \log_2 g \rfloor$, $\ulcorner\cdot\urcorner : \mathbb{Z}_g \to \{0,1\}^*$ a bit-encoding function, $[\cdot]_g : \{0,1\}^* \to \mathbb{Z}_g$ a *full domain hash* function ([BR96]), $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{\breve{g}}$ a collision-resistant hash function and $\mathsf{ECDSA}(E, \mathsf{H}) = (E, \mathsf{Pub}, \mathsf{Sig}_{E,\mathsf{H}}, \mathsf{Ver}_{E,\mathsf{H}})$ the corresponding ECDSA scheme. With the notation from Section 2.1 we define $\Omega := \{0,1\}^{\breve{g}}$, $\mathbb{O} := \mathbb{E}^{\mathsf{M}}$, $\mathbb{P} := \big(\mathbb{Z}_g \cup \{\bot\}\big)^{\mathsf{M}}$, $\mathbb{T} := \mathrm{Im}(\mathsf{Sig}_{E,\mathsf{H}})$, $\mathbb{B} := \mathbb{Z}_g$.

[2] Using ECDSA is also not required: we have created an instantiation based on Edx25519 (Appendix A); ECDSA is merely one that permits a concise description.

Instantiation 1 (*General instantiation with ECDSA*) *Let E be an elliptic curve for which the decisional Diffie-Hellman assumption (DDH) holds [Bon98], i. e. given αG and βG it is computationally infeasible to distinguish between $\alpha\beta G$ and γG with uniformly random $\alpha, \beta, \gamma \in \mathbb{Z}_g$ with probability better than $\frac{1}{2} - \epsilon(\lambda)$.*

Let $p_i := [\omega, \overbracket{i}]_g$ for $i \in \{1, \ldots, \mathsf{M}\}$ be private keys, $q_i := \mathsf{Pub}_E(p_i)$ and let then the private keys $p_{\mathsf{a}+1}, \ldots, p_M$ be explicitly dropped by the guardian in:

$$\mathsf{Commit}_{E,[\cdot]_g}(\mathsf{a}, \omega) := \Big\langle \overbrace{(q_1, \ldots, q_{\mathsf{M}})}^{=\vec{\mathsf{Q}}}, \overbrace{(p_1, \ldots, p_{\mathsf{a}}, \bot, \ldots, \bot)}^{=\vec{\mathsf{P}},\ \textit{length}\ \mathsf{M}} \Big\rangle \tag{11}$$

$$\mathsf{Attest}_{E,\mathsf{H}}(\mathsf{b}, \vec{\mathsf{Q}}, \vec{\mathsf{P}}) := \begin{cases} \mathsf{T}_{\mathsf{b}} := \mathsf{Sig}_{E,\mathsf{H}}\big(\mathsf{b}, \vec{\mathsf{P}}[\mathsf{b}]\big) & \textit{if } \vec{\mathsf{P}}[\mathsf{b}] \overset{?}{\neq} \bot \\ \bot & \textit{otherwise} \end{cases} \tag{12}$$

$$\mathsf{Verify}_{E,\mathsf{H}}(\mathsf{b}, \vec{\mathsf{Q}}, \mathsf{T}) := \mathsf{Ver}_{E,\mathsf{H}}(\mathsf{b}, \vec{\mathsf{Q}}[\mathsf{b}], \mathsf{T}) \tag{13}$$

$$\mathsf{Derive}_{E,[\cdot]_g}(\vec{\mathsf{Q}}, \vec{\mathsf{P}}, \omega) := \Big\langle (\beta * q_1, \ldots, \beta * q_{\mathsf{M}}), (\beta p_1, \ldots, \beta p_{\mathsf{a}}, \bot, \ldots, \bot), \beta \Big\rangle \tag{14}$$

$$\textit{with } \beta := [\omega]_g \textit{ and multiplication } \beta p_i \textit{ modulo } g$$

$$\mathsf{Compare}_E(\vec{\mathsf{Q}}, \vec{\mathsf{Q}}', \beta) := \begin{cases} 1 & \textit{if } (\beta * q_1, \ldots, \beta * q_{\mathsf{M}}) \overset{?}{=} (q'_1, \ldots, q'_{\mathsf{M}}) \\ 0 & \textit{otherwise} \end{cases} \tag{15}$$

Then we call the tuple

$$\mathsf{AgeVer}(\lambda, E, [\cdot]_g, \mathsf{H}) := \\ (\lambda, \mathsf{ECDSA}_{E,\mathsf{H}}, \mathsf{Commit}_{E,[\cdot]_g}, \mathsf{Attest}_{E,\mathsf{H}}, \mathsf{Verify}_{E,\mathsf{H}}, \mathsf{Derive}_{E,[\cdot]_g}, \mathsf{Compare}_E)$$

a general instantiation *of (1)-(5) .* ■

It is straightforward to verify that this instantiation meets our basic requirements 1–4.

4 Proofs of the security properties

In this section, we will prove that the instantiation 1 fulfills the challenging security requirements 5–7.

Theorem 1. $\mathsf{AgeVer}(\lambda, E, [\cdot]_g, \mathsf{H})$ *(Instantiation 1) satisfies the* nondisclosure of age *requirement (Requirement 5).*

Proof. Note that in $G_{\mathcal{A}}^{\mathsf{AgeCA}}(\lambda, \mathsf{m}, n)$ (Game 1) the adversary is provided with the commitments Q_i's and β_i's (in case $n > 1$) which are independent of m and the randomly chosen $\mathsf{a} \in \{\mathsf{m}, \ldots, \mathsf{M}\}$ for all $n \in \mathbb{N}^+$ according to the definitions of $\mathsf{Commit}_{E,\mathsf{H}}$ and Derive_E in (11) and (14).

Also, with respect to the provided attestations $\mathsf{T}_i = \mathsf{Attest}(\mathsf{m}, \mathsf{Q}_i, \mathsf{P}_i) = \mathsf{Sig}_{E,\mathsf{H}}(\mathsf{m}, \mathsf{P}_i[\mathsf{m}])$, m and $\mathsf{P}_i[\mathsf{m}]$ are independent of the randomly chosen $\mathsf{a} \in \{\mathsf{m}, \ldots, \mathsf{M}\}$, for all $i \in \{1, \ldots, n\}$ and all $n \in \mathbb{N}^+$.

Therefore the adversary can only guess the value of a with probability $\frac{1}{\mathsf{M}-\mathsf{m}+1}$, for any $n \in \mathbb{N}^+$. □

Theorem 2. $\mathsf{AgeVer}(\lambda, E, [\cdot]_g, \mathsf{H})$ *(Instantiation 1) satisfies the* unforgeability of minimum age *requirement (Requirement 6).*

Proof. In order for the game $G^{\mathsf{FA}}_{\mathcal{A}}(\lambda, n)$ (Game 2) to return 1, the ECDSA signature verification $(\mathsf{Verify}_{E,\mathsf{H}}(\mathsf{m}, \mathsf{Q}, \mathsf{T}) = \mathsf{Ver}_{E,\mathsf{H}}(\mathsf{m}, \mathsf{Q}[\mathsf{m}], \mathsf{T}))$ must be successful. Note that according to the definition (11) the adversary is initially not provided with the private keys $\{p_{\mathsf{a}+1}, \ldots, p_{\mathsf{M}}\}$ and subsequent calls to Derive do not yield those neither for $\mathsf{m} \in \{\mathsf{a}+1, \ldots, \mathsf{M}\}$. Winning this game, for any $n \in \mathbb{N}$, is therefore equivalent to existential forgery of ECDSA, which has negligible probability. □

Theorem 3. $\mathsf{AgeVer}(\lambda, E, [\cdot]_g, \mathsf{H})$ *(Instantiation 1) satisfies the* unlinkability of commitments and attestations *requirement (Requirement 7).*

Proof. First we show that the adversary gets no information out of the commitments Q^j_i, A^k_i and B^k_i in the game $G^{\mathsf{DCA}}_{\mathcal{A}_0,\mathcal{A}_1}(\lambda, \kappa, n)$ (Game 3). The DDH assumption of the elliptic curve extends to uniformly random vectors $(\alpha_1, \ldots, \alpha_{\mathsf{M}})$, $(\beta_1, \ldots, \beta_{\mathsf{M}}) \in \mathbb{Z}^{\mathsf{M}}_g$ and $\gamma \in \mathbb{Z}_g$: Given $(\alpha_1 G, \ldots, \alpha_{\mathsf{M}} G)$, $(\beta_1 G, \ldots, \beta_{\mathsf{M}} G) \in \mathbb{E}^{\mathsf{M}}$ and $\gamma G \in \mathbb{E}$ the vector of points $(\gamma\alpha_1 G, \ldots, \gamma\alpha_{\mathsf{M}} G)$ can be distinguished from $(\gamma\beta_1 G, \ldots, \gamma\beta_{\mathsf{M}} G)$ again only with probability $\frac{1}{2} - \epsilon(\lambda)$, absorbing the constant M. The use of the FDH $[\cdot]_g$ in $\mathsf{Commit}()$ and $\mathsf{Derive}()$ guarantees that all components of Q^j_i, A^k_i and B^k_i are uniformly distributed in $\mathbb{E}$.

Note that $\mathsf{Compare}()$ is not a distinguisher for the adversary, as it will only return 1 for Q^j_i and each of the commitments and bindings in the corresponding vector $\vec{R}^j_i$ *within* each triple $(\mathsf{Q}^j_i, T^j_i, \vec{R}^j_i)$. When provided with $\mathsf{Q}^b_{i_b}$ and any commitment and blinding from *both* histories, it returns 1 only with probability $\epsilon(\lambda)$ (non-zero due to the uniform distribution of the points on the finite elliptic curve).

Finally, the adversary is provided with $T^b_{i_b} = \mathsf{Sig}_{E,\mathsf{H}}(m_{i_b}, P^b_{i_b})$. Verify will only return 1 with the commitment from the same triplet, but for any other commitment in any of the two histories it will return 1 only with probability $\epsilon(\lambda)$ (again, non-zero due to the uniform distribution of the points on the finite elliptic curve). And because $m_{i_0} = m_{i_1}$ the adversary can not distinguish the indices.

Therefore, the adversary can only guess b correctly with probability $\frac{1}{2} - \epsilon(\lambda)$. □

5 Background: GNU Taler

GNU Taler is a token-based electronic online payment system using cryptography to secure payments. It extends the concepts introduced by eCash in 1989 [Cha89]. A coin in GNU Taler is a public/private key pair where the private key is only known to the owner of the coin. GNU Taler provides accountability and protect citizens' right to informational self-determination [Dol19] and can be used by commercial banks interested in underwriting commercial e-money, or as a central bank digital currency (CBDC) [CGM21]. GNU Taler meets—among others—the following security and privacy goals which are in alignment with the goals of our age restriction scheme:

1. Purchases must not identify the buyer, and must also not be linkable to other transactions of the same buyer.
2. Coins must be fungible. That is, all coins signed with the same denomination key must be equivalent. In particular, it must not be possible to partition the anonymity set into between users that used change and those that used cash that was directly withdrawn.
3. Customers must be always able to pay any amount for which they have sufficient total digital coins and receive change in an unlinkable way.

We will now summarize the key steps of the GNU Taler protocols that are relevant to our extension for age restriction. See also figure 2 for a schematic overview of the participants and the protocols between them. The complete protocol suite with all the details is defined in [Dol19].

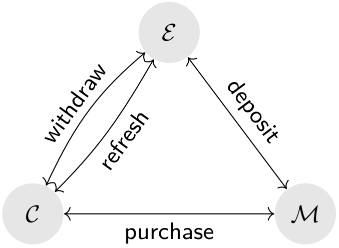


Fig. 2: *Overview of the GNU Taler protocols (partially)* – The customer $\mathcal{C}$ withdraws coins from the payment service provider (exchange) $\mathcal{E}$. $\mathcal{C}$ uses the coins to purchase at a merchant $\mathcal{M}$, who then deposits the coins at $\mathcal{E}$. $\mathcal{C}$ gets change for coins from $\mathcal{E}$ via the zero-knowledge refresh protocol.

withdraw [Dol19, §4.7.2]: For each coin, the customer $\mathcal{C}$ creates a pair (c_i, C_i) of private and public keys. $\mathcal{C}$ then requests the payment service provider $\mathcal{E}$ to create a blind signature over C_i using the private key to denomination D_v, which represents a particular unit of value, authorizing $\mathcal{E}$ to deduct the respective balance from the consumer's account. The blind signature is made over the full domain hash [BR96] of the public key C_i of the coin and the validation of a coin is performed by signature validation:

$$1 \overset{?}{=} \mathsf{SigCheck}\big(\mathsf{FDH}(C_i), D_v, \sigma_i\big) \tag{16}$$

Here, SigCheck is the verification function of the blind signature scheme[3], D_v is the public key of the denomination and σ_i is the signature to be verified.

purchase [Dol19, §4.7.3]: To pay for goods, $\mathcal{C}$ first negotiates a contract with the merchant $\mathcal{M}$. Upon agreement, the merchant $\mathcal{M}$ cryptographically signs the

[3] GNU Taler currently supports RSA [Cha89] and Clause Blind Schnorr [DH22; Ban21] blind signature schemes.

contract and $\mathcal{C}$—their identity possibly remaining private—signs the contract, too, with private coin keys c_i and sends it to $\mathcal{M}$.

deposit [Dol19, §4.7.3]: $\mathcal{M}$ forwards the signed contract to $\mathcal{E}$. The signatures, performed with valid coins c_i from $\mathcal{C}$, are basically instructions from $\mathcal{C}$ to $\mathcal{E}$ to pay the merchant $\mathcal{M}$ who is identified by the bank account details in the contract. $\mathcal{E}$ checks for overspending and the validity of each coin itself, given its public key C_i and using the signature verification with the formula in (16).

refresh [Dol19, §4.7.4]: $\mathcal{C}$ can ask $\mathcal{E}$ for change for a partially spend coin c_{old}. In order to maintain unlinkability of old and new coins, both parties perform a zero-knowledge, cut-and-choose protocol, with a security parameter $\kappa > 1$: $\mathcal{C}$ derives from C_{old} new coins $\{(c_1, C_1), \dots, (c_\kappa, C_\kappa)\}$ and sends $\mathcal{E}$ a commitment to $(\beta_1(C_1), \dots, \beta_\kappa(C_\kappa))$ without disclosing the C_i by using blinding functions β_i. $\mathcal{E}$ then chooses a $\gamma \in \{1, \dots, \kappa\}$ and $\mathcal{C}$ has to prove ownership of c_{old} and disclose the correct key derivation and the blindings β_i for all $i \neq \gamma$, which proves (with certainty $\frac{\kappa-1}{\kappa}$) the ownership by $\mathcal{C}$ of c_{old} and all but one c_i. Together with the blinded $\beta_\gamma(C_\gamma)$, $\mathcal{E}$ can compare the computed values with the initial commitment. On success, $\mathcal{C}$ receives a blind signature with the appropriate denomination for undisclosed fresh coin C_γ.

6 Integration into GNU Taler

We now present the integration of the the age-restriction scheme 2.1 into GNU Taler [Dol19, §4.7].

A crucial step is to indisputably bind a particular age commitment to a particular coin. This is done by requiring the blind signature of a coin's public key C_p in the original protocol to now also include the age commitment Q. Specifically, instead of signing $\mathsf{FDH}(C_p)$, the exchange will now blindly sign $\mathsf{FDH}(C_p, H(\mathsf{Q}))$. This means that instead of the original signature check in equation (16), now the check of validity of a coin's signature requires the hash of the age commitment:

$$1 \stackrel{?}{=} \mathsf{SigCheck}\big(\mathsf{FDH}(C_p, H(\mathsf{Q})), D_p, \sigma_p\big) \tag{17}$$

Again, $\mathsf{SigCheck}$ is the verification function of the signature scheme, D_p is the public key of the denomination and σ_p is the signature to be verified.

With the tight bond between a coin's public key and an age commitment defined, the existing procotols from GNU Taler are augmented as follows (see figure 3 for a schematic overview):

$\mathsf{withdraw} \mapsto (\mathsf{Commit}, \mathsf{withdraw})$**:** A guardian $\mathcal{G}$ Commit to an age a, producing a commitment Q and a proof P_a.The commitment Q is bound to a fresh coin's public key C_p during the $\mathsf{withdraw}$ protocol by generating a blind signature for $\mathsf{FDH}(C_p, H(\mathsf{Q}))$ (instead of $\mathsf{FDH}(C_p)$ as in the original protocol).

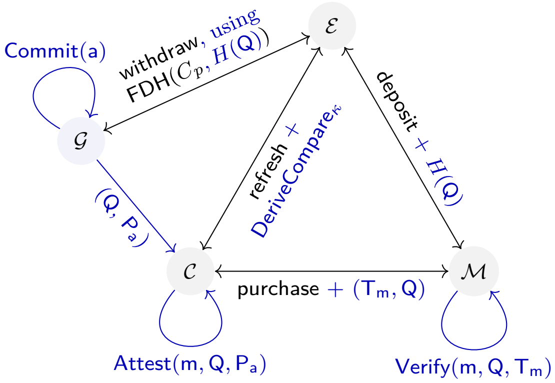


Fig. 3: *Extended Taler protocol suite* – The protocols withdraw, purchase, deposit and refresh from figure 2 and call graphs to Commit, Attest, Verify from figure 1, together with the cut&choose protocol DeriveCompare$_\kappa$, defined in Section 2.2, are combined into a suite of augmented protocols for GNU Taler.

purchase ↦ (Attest, purchase, Verify): A merchant $\mathcal{M}$ can specify a minimal age m as requirement for a purchase in the contract terms. Assuming that $\mathsf{m} \le \mathsf{a}$, the child can now Attest the minimum age and send the attestation $\mathsf{T_m}$ and commitment Q to the merchant during the purchase protocol. The merchant can Verify the minimum age m, given $\mathsf{T_m}$ and Q. Note that the merchant can also verify that the commitment Q was bound to the coin C_p by verifying the signature of $\mathsf{FDH}(C_p, H(\mathsf{Q}))$.

deposit: Within the deposit protocol, $\mathcal{M}$ now also sends $H(\mathsf{Q})$ to the payment service provider $\mathcal{E}$, who can then verify the coin by verifying the signature of $\mathsf{FDH}(C_p, H(\mathsf{Q}))$.

refresh ↦ (DeriveCompare$_\kappa$, refresh): The two cut-and-choose protocols, the original refresh protocol and the age restriction specific DeriveCompare$_\kappa$ from section 2.2, are run in parallel with the same $\gamma \in \{1, \dots, \kappa\}$ chosen by $\mathcal{E}$. However, instead of sending (blinded) $\mathsf{FDH}(C_i)$ for the original refresh protocol, $\mathcal{C}$ sends (the blinded) $\mathsf{FDH}(C_i, H(\mathsf{Q}_i))$. When both, refresh and DeriveCompare$_\kappa$, terminate successfully, $\mathcal{E}$ blindly signs $\mathsf{FDH}(C_\gamma, H(\mathsf{Q}_\gamma))$.

7 Implementation and Benchmarks

Table 1 summarizes the performance of our five key operations. We note that the time the wallet spends on computing Commit and Derive is largely insignificant for the user experience as it happens during non-interactive background operations. Similarly, the latency from Attest can typically be hidden by precomputing the result while the human user is busy reviewing the terms of the sale and making the purchasing decision. The latency increase from an exchange computing Compare is not relevant for the user experience as it again happens during a non-interactive background operation. However, it may require adequate provisioning of computational resources at the exchange. Only Verify is crucial for the user experience, as it runs at the merchant on the critical path between the user confirming the payment and the ultimate payment confirmation and fulfillment. Fortunately, this is also a cheap operation for the merchant.

	Impl.	Commit	Attest	Verify	Derive	Compare
ECDSA	Go-AMD[1]	209.6± 101.5	69.2± 28.1	125.9± 66.9	775.8± 314.0	603.7± 230.8
	Go-PEN[2]	322.8± 14.0	83.5± 11.5	218.6± 64.1	1579.0± 84.7	1292.0± 52.1
	Go-ARM[3]	6097.0± 47.6	1073.0±128.1	2856.0± 21.9	21309.0± 53.9	15901.0± 46.8
	C-AMD[1]	1741.9± 49.1	445.7± 29.3	610.4± 9.8	5523.5± 68.0	– [5]
Edx25519	Go-AMD[1]	219.7± 78.3	70.3± 28.0	94.4± 33.5	1158.0± 564.2	885.9± 411.0
	Go-PEN[2]	395.8± 19.1	139.1± 9.6	190.1± 12.4	2053.0± 105.5	1536.0± 74.9
	Go-ARM[3]	3311.0± 31.8	1213.0± 13.5	1870.0± 16.6	18006.0± 61.5	14017.0± 603.4
	C-AMD[1]	272.9± 61.1	48.7± 5.4	72.1± 7.0	4948.6± 37.0	– [5]
	C-PEN[2]	433.0± 30.6	113.5± 12.0	174.0± 8.9	3882.6± 89.2	– [5]
	TS-i7[4]	50412.5±5459.7	5882.9±692.0	11095.2±1007.2	131728.4±5376.0	88060.1±4662.3

Table 1: Runtime in µs (average and standard deviation of min. 500 iterations) of the various operations for eight age groups ($M = 8$) implemented in different languages and run on various CPUs (single-core). The ECDSA implementation in Go uses curve NIST-256 instead of curve25519 due to the missing full support for all arithmetic operations. [1]) AMD Ryzen 7 3750H. [2]) Intel Pentium Silver N5000 CPU 1.10GHz. [3]) ARM-Cortex-A72 1.8GHz. [4]) TypeScript on Intel Core i7-10510U CPU. [5]) The C-implementation for $\mathsf{DeriveCompare}_\kappa$ has an optimization in which the call to $\mathsf{Compare}$ is not necessary and therefore not implemented.

The additional bandwidth required for a regular withdraw and deposit is 32 bytes for the additional transmission of $H(\mathsf{Q})$. During the purchase, an additional $(M + 1) \cdot 32$ bytes need to be transmitted for $\mathsf{T_m}$ and Q. This additional information may also need to be stored by the exchange and merchant for many years to enable later audits.

We have implemented our protocol in the GNU Taler system, specifically the Taler exchange, merchant and wallet components. As part of GNU Taler our implementation is free software under AGPL. For the current implementation in GNU Taler, we are using Edx25519 (Appendix A) for a security level of 128 bits.

8 Discussion

GNU Taler — Our design for age restriction protocol is quite general and can be instantiated with various cryptographic primitives. It can also in principle be used with any token-based payment service. However, its design goals—in terms of security, privacy and efficiency—and participants strongly align with those of GNU Taler.

Minors with bank accounts — One key principle in our design is subsidiarity: Wards or parents are the entities that effectively choose the age limits for the coins for their wards. This hinges on the assumption that personal bank accounts are owned by adults. In countries where also minors have personal bank accounts, minors could withdraw digital cash without an adult ensuring that an age restriction is set by the wallet at the time of withdrawal. To address this case, we assume that banks provide the Taler exchange with the minimum age of the minor whenever a debited account is age-restricted. This would typically happen whenever a Taler reserve is credited by a wire transfer from a minor's bank account. To ensure that withdrawn coins are still created with an age restriction in this case, a variant of the cut-and-choose approach of the extended refresh protocol can be used. But, instead of proving the equivalence of the age restrictions, the minor would prove (with probability $\frac{\kappa-1}{\kappa}$) that the Q included public keys for which the minor *does not know the private key* at the slots corresponding to restricted age levels. This can be achieved by proving that these commitments were derived from a well-known master public key of the system.[4] To ensure the resulting coins are indistinguishable from all other coins even when the Q is disclosed to the merchant, the protocol must use different β values for each derivation from the master public key. So here Derive would need to be adjusted to operate on each element instead of a vector.

Limitations — In our design wards get to determine the appropriate age-level, which can differ from the biological age in both directions. The design also does not handle the case where the individual gets older, leaving it to both parties to negotiate how to proceed. In any case, a wallet implementation for GNU Taler should prioritize using coins with lower age-restrictions, and coins in Taler are meant for spending and not for hoarding.

The protocols of GNU Taler itself are not post-quantum (PQ) ready, as we do not know of any suitable PQ blind signature scheme.

Identity management systems — Age is an important part of a persons identity, and handling identity information requires high standards of protection and confidentiality and raises sensitive ethical questions. Here we want to discuss some problems, for which digital identity systems are not the only and often not the best solution:

- Some proposals to replace cash for central bank digital currencies [MT19] introduce digital identities to discharge KYC requirements needed for retail central bank accounts [Int21].

  GNU Taler demonstrates that *anonymous* digital cash is feasible and proposes a two-tiered architecture where central banks can satisfy regulatory requirements at scale by piggy-backing on existing commercial bank processes.

- Mastercard's "Trust Stamp" [Fau20] project intends to link vaccination data with personal biometric data to create a digital health passport in the context of the GAVI alliance, with critics pointing out [Bea21] the potential for

[4] The private key of the master public key must simply be deleted after creation, as it would enable minors to defeat the cut-and-choose protocol. Deriving commitments from the master key implies that computing the private key corresponding to the commitment is equivalent of solving DLOG for the master public key.

abuse by AI-powered predictive policing of the biometrically tagged population.

D3PT [Tro+20] and Europe's vaccination certificates [Ede21] demonstrate that more decentralized and privacy-respecting approaches are viable alternatives to fight pandemics.

- Online protections for minors are another area where digital passports have been proposed as a solution by surveillance-friendly governments [Her21].

  The protocol presented in this paper provides a method for protecting minors where the state only sets the rules for commercial providers, while leaving the actual decisions to the minor's wards — where it belongs. [Bos10]

Identity bases systems are also not very popular: In a recent election, the Swiss population rejected the creation of a public-private partnership for digital identity management [Bun21], despite digital identity systems being proposed to the voters as a solution for many social problems.

9 Related Work

To our knowledge, all currently available systems for privacy-preserving age restrictions are based on attribute-based credentials [Kon+14; CL01; CDL16; Sch+19; Au+12], where authorities issue consumers with a certificate that enables them to produce a zero-knowledge proof [GMR89] showing that they are of sufficient age. This identity-centric approach is also reflected in emerging standards for self-sovereign identity [Con+19].

However, in order for identity providers to issue statements as attribute-based credentials from their respective subjects, they are implicitly expected to collect and verify the respective personal information. Critically, our approach does not require the existence of a dedicated identity provider and instead relies on the principle of subsidiarity as part of the payment system.

This attribute-based approach lacks broad deployment mainly for two reasons: First, it remains complex for consumers and retailers, and second, it requires authorities to issue suitable credentials even for self-sovereign identity systems [Sch+21]. The complexity arises from fundamental open questions of trust in context of self-sovereign identity. Which authorities can or should be trusted with attesting user information? Is it reasonable to assume that this information can be protected appropriately by the identity provider? The principle of subsidiarity as integrated in our approach offers an elegant solution to this conundrum by completely sidestepping questions of identity and trust.

Other approaches which tie age restriction to the ability to pay do exist. For example, specialized credit cards for children limit the ability to pay to certain "safe" stores. [Fea21] This approach has the advantage that the age restriction is part of the mandatory payment process. Hence, consumers do not have to perform additional steps during checkout. This is crucial as additional steps during checkout are problematic for retailers because they increase costs and may even lead to consumers aborting the purchase process. We argue that while age restriction as a feature of the payment system is clearly desirable, the existing credit card process is not privacy-friendly: They require minors to register with payment service providers which can than identify and track purchases of minors. Furthermore, restricting payments to specific stores is also unnecessarily restrictive.

10 Conclusion

Age restriction in e-commerce is not merely a technical challenge. It is a matter of ethical and legal origin for which, so far, only technological solutions without strong protection of privacy or solutions based on identity management systems exists.

Our work thus contributes to the technological solution space by providing a privacy-friendly age restriction scheme based on subsidiarity. It adds to a body of research that questions the basis on which policy makers justify the deployment of identity management systems.

A Edx25519

Edx25519 is a signature scheme based on Ed25519 [Ber+12], but allows for derivation of private and public keys, independently, from existing ones. Private keys in Edx25519 are pairs (a, b) of 32 byte each. Initially they correspond to the result of the expansion and clamping in EdDSA. The scheme is as follows, in pseudo-code:

```
Edx25519_generate_private(seed) {
  // EdDSA expand and clamp
  dh := SHA-512(seed)
  a := dh[0..31]
  b := dh[32..64]
  a[00] &= 0b11111000
  a[31] &= 0b00111111
  a[31] |= 0b01000000
  return (a, b)
}

Edx25519_public_from_private(private) {
  // Public keys are the same as in EdDSA
  (a, _) := private
  return [a] * G
}

Edx25519_sign(private, message) {
  // Identical to Ed25519, except for the origin of b
  (a, b) := private
  P := Edx25519_public_from_private(private)
  r := SHA-512(b ‖ message)
  R := [r] * G
  s := r + SHA-512(R ‖ P ‖ message) * a % L
  return (R,s)
}

Edx25519_verify(P, message, signature) {
  // Identical to Ed25519
  (R, s) := signature
  return [s] * G == R + [SHA-512(R ‖ P ‖ message)] * P
}

Edx25519_blinding_factor(P, seed) {
  // This is a helper function used in the derivation of
  // private/public keys from existing ones.
  h1 := HKDF_32(P, seed)
  // Ensure that h == h % L
  h  := h1 % L
  // Make sure that we don't create weak keys.
  P' := [h] * P
  if !( (h!=1) && (h!=0) && (P'!=E) ) {
    throw error
  }
  return h
}

Edx25519_derive_private(private, seed) {
  (a, b) := private
  P := Edx25519_public_key_from_private(private)
  h := Edx25519_blinding_factor(P, seed)
  // Carefully calculate the new value for a
  a1 := a / 8;
  a2 := (h  * a1) % L
  a' := (a2 *  8) % L
  // Update b as well, binding it to h.
  b' := SHA256(b ‖ h)
  return (a', b')
}

Edx25519_derive_public(P, seed) {
  h := Edx25519_blinding_factor(P, seed)
  return [h]*P
}
```

References

[Cha89] David Chaum. “Blind Signatures for Untraceable Payments”. In: *Advances in Cryptology* Proceedings of Crypto82 (1989). DOI: 10.1007/978-1-4757-0602-4_18.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208. DOI: 10.1137/0218012.

[BR96] Mihir Bellare and Phillip Rogaway. "The Exact Security of Digital Signatures - How to Sign with RSA and Rabin". In: *Advances in Cryptology - EUROCRYPT '96*. Vol. 1070. LNCS. Springer, 1996, pp. 399–416. DOI: 10.1007/3-540-68339-9_34.

[Bon98] Dan Boneh. "The Decision Diffie-Hellman Problem". In: *Third Algorithmic Number Theory Symposium*. Vol. 1423. Springer-Verlag, 1998, pp. 48–63. DOI: 10.1007/BFb0054851.

[CL01] Jan Camenisch and Anna Lysyanskaya. "An efficient system for non-transferable anonymous credentials with optional anonymity revocation". In: *Int. conf. on the theory and applications of cryptographic techniques*. Springer. 2001, pp. 93–118.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. "The Elliptic Curve Digital Signature Algorithm (ECDSA)". In: *International journal of information security* 1.1 (2001), pp. 36–63. DOI: 10.1007/s102070100002.

[Bos10] David A. Bosnich. "The Principle of Subsidiarity". In: *Religion & Liberty* 6.4 (July 2010).

[Pou11] Yves Poullet. "e-Youth before its judges — Legal protection of minors in cyberspace". In: *Computer Law & Security Review* 27.1 (2011), pp. 6–20. ISSN: 0267-3649. DOI: 10.1016/j.clsr.2010.11.011.

[Au+12] Man Ho Au et al. "Constant-size dynamic k-times anonymous authentication". In: *IEEE Systems Journal* 7.2 (2012), pp. 249–261.

[Ber+12] D.J. Bernstein et al. "High-speed high-security signatures". In: *J Cryptogr Eng* 2 (2012), pp. 77–89. DOI: 10.1007/s13389-012-0027-1.

[Kon+14] Merel Koning et al. "The ABC of ABC: An analysis of attribute-based credentials in the light of data protection, privacy and identity". In: *Proc. 10th International Conf. on Internet, Law & Politics* (2014), pp. 357–374.

[CDL16] Jan Camenisch, Manu Drijvers, and Anja Lehmann. "Anonymous attestation using the strong diffie hellman assumption revisited". In: *International Conf. on Trust and Trustworthy Computing*. Springer. 2016, pp. 1–20.

[Con+19] World Wide Web Consortium et al. "Verifiable credentials data model 1.0: Expressing verifiable information on the web". In: (2019). URL: https://www.w3.org/TR/vc-data-model/?#core-data-model.

[Dol19] Florian Dold. "GNU Taler - Practical and Provably Secure Electronic Payments". PhD thesis. 2019. URL: https://taler.net/papers/thesis-dold-phd-2019.pdf.

[MT19] Rodrigo Mejía-Ricart and Camilo Tellez-Merchan. *Distributed Ledger Technology and Digital Identity: Prospects and Pitfalls Ahead*. https://www.betterthancash.org/news/distributed-ledger-technology-

and-digital-identity-prospects-and-pitfalls-ahead. June 2019.

[Sch+19] Martin Schanzenbach et al. "ZKlaims: Privacy-preserving Attribute-based Credentials using Non-interactive Zero-knowledge Techniques". In: *Proc. 16th International Joint Conf. on e-Business and Telecom.* (2019). DOI: 10.5220/0007772903250332.

[Fau20] Miriam Fauzia. "Fact check: Mastercard's partnership on vaccination records is unrelated to finances". In: *USA Today* (2020).

[Tro+20] Carmela Troncoso et al. *Decentralized Privacy-Preserving Proximity Tracing.* Tech. rep. EPFL, 2020.

[Ban21] Aritra Banerjee. *A Fully Anonymous e-Voting Protocol Employing Universal zk-SNARKs and Smart Contracts.* Cryptology ePrint Archive, Report 2021/877. https://ia.cr/2021/877. 2021.

[Bea21] The Liberty Beacon. *'Trust Stamp' Vaccine Record And Payment System To Be Tested On Low-Income Africans.* Feb. 2021. URL: https://www.thelibertybeacon.com/trust-stamp-vaccine-record-and-payment-system-to-be-tested-on-low-income-africans/.

[Bun21] Bundeskanzlei. *Vorlage Nr. 639: Resultate in den Kantonen.* https://www.bk.admin.ch/ch/d/pore/va/20210307/can639.html. Mar. 2021.

[CGM21] David Chaum, Christian Grothoff, and Thomas Moser. "How to issue a central bank digital currency". In: *SNB working paper seires* (2021). URL: https://www.snb.ch/en/mmr/papers/id/working_paper_2021_03.

[Ede21] Daniel Eder. *EU Digital COVID Certificates Project.* https://github.com/eu-digital-green-certificates. June 2021.

[Fea21] Todd Feathers. "Debit Card Apps for Kids Are Collecting a Shocking Amount of Personal Data". In: *Motherboard* (July 2021).

[Her21] Alex Hern. "Can facial analysis technology create a child-safe internet?" In: *The Guardian* (July 2021).

[Int21] Bank of International Settlement. *Central bank digital currencies herald a new chapter for the monetary system.* https://www.bis.org/press/p210623.htm. June 2021.

[Pav21] Eeva Pavy. *The principle of subsidiarity.* https://www.europarl.europa.eu/factsheets/en/sheet/7/the-principle-of-subsidiarity. 2021.

[Sch+21] Martin Schanzenbach et al. "Decentralized Identities for Self-sovereign End-users (DISSENS)". In: *Open Identity Summit.* Gesellschaft für Informatik, 2021.

[DH22] Gian Demarmels and Lucien Heuzeveldt. "Adding Schnorr's Blind Signature in Taler". https://taler.net/papers/cs-thesis.pdf. Bacherlor's thesis. 2022.