Privacy Coins Under Viewing Key Compromise

Adrian Cinal[0009−0001−9789−0470]

NASK National Research Institute
adrian.cinal@nask.pl

Abstract. Anonymity guarantees of privacy-oriented cryptocurrencies are garnering negative attention from lawmakers who view them as antinomic to accountability. Having recognized their potential for innovation, however, regulators may not want to outright ban privacy coins but instead seek a middle ground where financial oversight is effective, and still some privacy is maintained. Mature designs, such as Zcash, Monero, or Firo, are expected to facilitate this through so-called *viewing keys* that can be disclosed to third parties for the purpose of supervision. This paper studies which privacy guarantees continue to hold once they have been. In doing so, it fills the gap in provable anonymity guarantees for Zcash and Firo under the compromise of the incoming viewing key, but nevertheless shows that malicious parties may happily surrender *all* their viewing keys and still find very efficient ways of evading financial monitoring. Of independent interest are several shortcomings identified in the prevailing security arguments for Monero.

Keywords: Privacy coins · Compliance · Anamorphic cryptography

1 Introduction

Privacy enhancement is a delicate balancing act with exercising a fundamental human right on the one hand and, on the other, equipping malicious actors with powerful tools for evading justice [15]. With technology capable of granting anonymity indiscriminately to honest users and criminals alike, the increased attention legislators have been paying cryptography in recent years is justified. Of most concern, naturally, are technologies involving financial instruments such as cryptocurrencies as a domain where accountability and transparency are most desired by state actors, *e.g.*, a revenue service, and, at the same time, incentives to be dishonest are the most tangible. This concern is reflected in, *e.g.*, recent European Union legislation [69, 70] as well as various other rulings around the world [61]. As Bitcoin is now well understood to not provide any meaningful anonymity [7, 51, 63, 66], one sees a transition to more technologically sophisticated cryptocurrencies, so-called "privacy coins," that offer stronger guarantees. This can be observed both among regular users[1] [42] and criminals [20,21,36,65].

[1] For example, Zcash saw a nearly twentyfold increase in price between year 2025 and 2026. See `https://coinmarketcap.com` (accessed: 30 August 2026).

With transaction histories cryptographically hidden, blockchain intelligence services and coin blacklisting [3,5,6,53,56], both standard tools for introducing accountability to cryptocurrencies, become ineffective against privacy coins. So do KYC practices at the crypto-fiat boundary [55]. In light of this, legislators may either rush to prohibition (see, *e.g.*, [70, Article 76(3)] or [71, Article 79(1)]), risking driving usage off regulated rails and worsening visibility, or recognize their potential for good (*e.g.*, protection of individual privacy, shielding businesses from corporate espionage, *etc.*) and search for new avenues of financial oversight in the technical landscape of privacy coins.

Mature designs, such as Monero [46, 64] or Zcash [11, 40],[2] take note of the compliance obligations and incorporate mechanisms that facilitate some degree of oversight through *viewing keys*, trapdoors that can be disclosed to a supervising authority to grant them varying levels of insight into the subject's activity. They can, *e.g.*, enable decryption of incoming transactions or recognizing when some asset gets spent. Viewing keys are commonly expected to play some role in financial monitoring [26, 27, 33, 58]. For example, before withdrawing to an external wallet, clients could be required to surrender the corresponding viewing keys. This could be stipulated in the exchange's terms of service under, *e.g.*, anti-money laundering (AML) law (*cf.* [67, 69, 71]). Also, the exchange may itself be obligated to submit its transactions to an audit, *e.g.*, for tax reporting purposes. The auditor could then demand viewing keys to ensure all transfers are correctly accounted for [10, 22, 25]. Little attention has been paid in the literature, however, to the privacy guarantees that are upheld in the case viewing keys are compromised. Also, the exact function they are to serve in financial monitoring is vaguely defined, and their effectiveness as a compliance tool has never been evaluated. These gaps are addressed in the present paper. In particular, if viewing key disclosure is to be mandated, it is crucial to (1) formally show viewing keys do not "overstep their boundaries" and indeed grant only the capability they were designed to grant, nothing more, as well as (2) evaluate to what degree bad actors can nonetheless evade oversight built around them. Notably, the former is of independent interest, as viewing keys are already routinely disclosed to untrusted servers to lift from lightweight wallets the burden of scanning the blockchain for inbound transfers [74]. The latter, in turn, is reminiscent of *anamorphic cryptography* [48, 49, 59], an analogy first observed in a recent case study of Monero [25]. Indeed, anamorphic cryptography is a field concerned with "possible abuses, misuses, and new uses of cryptosystems beyond their primary goals" [49] in the presence of a *dictator* who may demand participants in the system give up their private keys. We assume the financial regulator, just as the dictator in anamorphic cryptography, is not qualified to establish a new protocol design that would admit the desired balance of accountability and privacy (see, *e.g.*, [28,38]) but instead endorses an existing one (*cf.* [5,6,10,25,56]). Indeed, cryptocurrencies have great inertia to them and outcompeting existing solutions may often not be feasible due to strong network effects, so it is "im-

[2] Capitalized at, respectively, 8.84 and 14.12 billion US dollars as of this writing according to `https://coinmarketcap.com` (accessed: 30 August 2026).

portant to explore the practical options for taming the systems that already exist" [5]. To use the language of Chatzigiannis *et al.* [22], we shall therefore mostly (except briefly in Section 4.2) be interested in *auditability functionalities* as opposed to *accountability policies* which would necessarily entail changes to the consensus layer (*cf.* [22, Insight 6]). Finally, anamorphic cryptography strives for plausible deniability and assumes a "rational dictator" that will not prosecute their subjects groundlessly. Likewise, free societies usually require that suspicion be established before conducting an investigation, lest it be deemed unlawful and any evidence collected become inadmissible in court [30]. In light of the above similarities between financial monitoring under viewing key disclosure and anamorphic cryptography, the present paper will consistently use the term *dictator* even when dealing with, say, an auditor in a "democratic" setting. Similarly, for clarity, it will refer to their auditees as *subjects*.

1.1 Paper Contributions

The paper evaluates which privacy coin guarantees continue to hold in the case viewing keys are surrendered to a supervising authority (the dictator) and the limitations of such disclosure in enforcing accountability. To this end, it (1) formalizes in Section 3 spend indistinguishability relative to a set of viewing keys disclosed, and, in particular, (2) studies whether it holds (as it is implicitly expected to by the users) relative to standard so-called "incoming" viewing keys in existing privacy coins. (3) Zcash and Firo are shown to satisfy it, but several problems in proving this for Monero are identified, which were so far overlooked. Specifically, we (4) argue that, if UTXOs are minted honestly, then the public algebraic structure of transactions precludes any black-box reduction from any anonymity claim of any underlying ring signature scheme. Of independent interest is that we also (5) find and fix an error in the particular anonymity experiment of the CLSAG ring signature [39] deployed in Monero today. In Section 4, we (6) turn to evaluating the effectiveness of viewing key disclosure as a means for financial monitoring and demonstrate that even the disclosure of the so-called "full" viewing key is ineffective against sophisticated subjects. They may minimize the data they publish on chain to only what is strictly required by the consensus protocol and rely on off-chain (or subliminal) channels otherwise. Finally, we (7) propose how the issue can be remedied in a minimally intrusive fashion and while keeping data stored on chain to the absolute minimum.

2 Preliminaries

2.1 Notation

Let λ denote a security parameter. A function of λ is said to be *negligible* if it decays faster than the inverse of any polynomial in λ. An algorithm is said to be *efficient* if it is probabilistic and runs in time polynomial in the size of its input (which typically also means polynomial in λ). Write $x \leftarrow\!\!\$\ X$ to denote x being

sampled from a probability distribution X, *e.g.*, outputs of a probabilistic algorithm. When X is a set, it is understood that x is sampled uniformly at random from X. For consistency with reference literature, the notation for different keys used in Monero, Zcash, and Firo is based on, respectively, [46], [40] and [43], and introduced, as needed, below.

2.2 Privacy Coins

Cryptocurrencies rely on a distributed append-only ledger (a blockchain) to record transactions. For concreteness, this paper restricts its attention to cryptocurrencies based on the *unspent transaction output* (UTXO) model originally established by Bitcoin. This captures all practically relevant privacy coins.[3] Transactions in the UTXO model have outputs referred to as *notes* or, if still unspent, *UTXO*s. A transaction then *spends* (consumes) one or more existing (heretofore unspent) notes as well as *mints* (creates) one or more new notes (UTXOs). Notes are associated with amounts, and a transaction is valid (hence accepted by the network and eventually appended to the ledger) if the total amount minted is less than or equal to the total amount spent (no money is created), the inputs have not been spent before, and the creator proves to have been the recipient of the input notes. Normally, in, say, Bitcoin, the transaction ledger is fully transparent with transactions stored in plaintext. This is to the detriment of the privacy of the transacting parties [7, 29, 51, 63, 66]. To address this and enhance the anonymity and confidentiality of the users, privacy coins employ various cryptographic techniques.

In a privacy coin, details of a note, such as its amount or recipient's address, are hidden using cryptographic commitments. The consistency of a transaction is then proved in zero knowledge. This includes showing that the input and output amounts balance and that the originator has spend authority over the input being spent. The links between mints and spends are also hidden using various techniques such as ring signatures [37, 43, 46, 64] or zero-knowledge proofs of membership in a Merkle tree [11,40]. To prevent double-spending in this setting, *i.e.*, referencing the same output twice, notes have *tags* (called *key images* in Monero [46] and *nullifiers* in Zcash [40]) that must be revealed upon spending, an idea borrowed from Chaum's e-cash [23]. A tag is derived as a pseudorandom function of the note, and the set of tags revealed so far is maintained by the nodes of the network. If a tag is ever repeated, the offending transaction is rejected.

Proving the consistency of transactions may require some extra *witnesses* associated with each input note u being spent. These include, *e.g.*, the amount attached to u or any commitment trapdoors. These are normally communicated in band, namely, the creator of the note encrypts them under the recipient's public key and attaches the ciphertext to the note. This has the advantage that no off-chain communication or storage is needed, and the ledger is "self-contained."

[3] The UTXO model is overall more accommodating of privacy-enhancing technologies [8,17]. If need be, however, the spend indistinguishability notion from Section 3 can be easily modified to apply to the account-based setting as well.

A privacy-coin note u then looks as follows (compare with Appendix B):

$$(\text{note } u) = (\text{commitments } C) + (\text{encrypted witnesses } c)\,.$$

From now on, as shorthand, we will say "decrypt a note u" when we really mean "decrypt the ciphertext c found in u." Importantly for the present work, the consistency of the ciphertexts published is *not* subject to the consensus mechanism, *i.e.*, the network does not verify it in any way. Users are free to mint "malformed" notes with, say, c being completely random. They may then send any witnesses out of band to the recipient.

Normally, privacy coins have every user continually *scan* the blockchain looking for notes they can successfully decrypt. In mature constructions like Monero [46], Zcash [40], or Firo [43], this computational effort can typically be offloaded by the cryptocurrency wallet (say, a smartphone app) to a more capable machine by the use of viewing keys [74]. Indeed, instead of using just a single key for everything, privacy coins separate responsibilities so that different keys are used to, *e.g.*, decrypt the note, compute the tag, and prove spend authority. The key used to prove note ownership (spend authority) is typically called the *spending key* with all other keys being various *viewing keys*. Users may choose (or be compelled to) disclose viewing keys to a third party without granting them control over their assets. Importantly, besides facilitating lightweight clients, viewing keys are often proposed by the community as a means of financial monitoring [26, 27, 33, 58], a proposal the present work sets out to evaluate. The expectation is that the dictator, in control of their subjects' viewing keys, should get extra insight into the subjects' activities and be able to at least recognize incoming notes. They could then realistically demand the subjects reveal the details of these notes. For this purpose, all three of Monero, Zcash, and Firo define *incoming viewing keys* (denoted, respectively, k^v, ivk, and $\mathsf{addr}_{\mathsf{in}}$ in the literature), which are studied in Section 3. The viewing key hierarchy can get more nuanced, as is the case with Zcash and Firo, which define *full viewing keys* that are used in the computation of tags. Supposedly, a dictator in the possession of these should be able to account for all inbound and outgoing transactions having to do with a particular subject. What this paper shows in Section 4, however, is that even full viewing keys are only effective when dealing with "semi-honest" subjects that use some default implementation of wallet software and not rely on out-of-band (or subliminal [4,13,25]) channels. Finally, Zcash also defines an *outgoing viewing key* ovk that can (at the discretion of the transaction originator) enable its holder to decrypt the output note and learn the intended recipient. This is again not enforced by consensus [40, Section 4.20.1].

Since viewing keys play a crucial role in this work, privacy coins that do not define any, such as Quisquis [37, 57] or Mimblewimble [60], shall not be of any interest.[4] Also, for lack of space, a more detailed introduction of Monero, Zcash, and Firo is relegated to Appendix B. Comprehensive descriptions of the full protocols can be found in [46], [40], and [43], respectively.

[4] The Omniring paper [50] mentions the possibility of extending it with viewing keys, but the details are not elaborated.

3 Spend Indistinguishability Relative to Viewing Keys

The most practical way to impose financial oversight on privacy-oriented cryptocurrencies, in particular those already deployed, is to obligate the subjects legally (but not technically) to submit their transactions for "clearing" with the dictator. The subjects would then have to reveal and attest to the information requested by the dictator such as the recipient or the amount [10, 18, 25]. Obviously, bad actors will not play by the rules and instead leverage the cryptographic guarantees of a privacy coin to hide the money flow. Viewing keys are often proposed as a means for enforcing accountability even in this setting [26, 27, 33, 58]. For example, cryptocurrency exchanges may be monitored (by, *e.g.*, a financial supervision authority) for deposits from self-hosted wallets, justifying mandating the disclosure of the incoming viewing key [25, 69]. While this enables the dictator to identify incoming funds (*i.e.*, "credits" to the exchange), they intuitively *should* remain oblivious to outgoing transfers: disclosure of the incoming viewing key is not expected to violate sender anonymity. This becomes even more of an issue when one considers another common reason to disclose incoming viewing keys, namely, scanning the ledger for inbound transfers, which is a responsibility that lightweight clients normally offload to an untrusted server. Both the above concerns: (1) detecting failure to submit a transaction to an audit and (2) not overstepping the boundaries of viewing key disclosure are captured in a unified model in this section.

The literature so far has been lacking in the evaluation of privacy coins relative to different viewing keys that the dictator can compel their subjects to disclose. Zcash Improvement Proposal number 310 [41] lists informal guarantees for the dictator regarding the consistency of the view they obtain when given the subject's full viewing key fvk. Guarantees relative to "weaker" viewing keys down the Zcash key hierarchy [40, Section 3.1] are not considered there.[5] What is more alarming still is that there seems to be a non-trivial gap to be filled in the provable security of Monero under viewing key compromise. Indeed, in the case viewing keys are known to the dictator, sender anonymity no longer follows from chosen-target linkable anonymity of the underlying ring signature scheme [39, Definition 9] (see Definition 3 in Appendix C). We reformulate the implicit belief of the community as Conjecture 1 and elaborate on the difficulty in resolving it via a black-box reduction from ring signature security. More generally, in the cases of both Monero and Firo (for which the corresponding proof has also been missing so far), viewing keys seem to be an afterthought added on top of an otherwise sound design without redoing the analysis.

3.1 Spend Indistinguishability Model

For the purposes of this paper, one must distinguish between UTXOs (more generally: notes) as perceived by the network and UTXOs as perceived by their

[5] The proposal is now also outdated, not reflecting the newest version of the Zcash protocol, Orchard (Ironwood), that is of interest to this paper; see `https://github.com/zcash/zips/issues/606` (accessed: 30 August 2026).

owners. Specifically, for each UTXO u (*i.e.*, a transaction output that has not been referenced before; see Appendix B for details), define the corresponding *owned UTXO* $\bar{u}$ which, in addition to u, contains all the private keys associated with u and witnesses necessary to spend u (compare this to the concept of an *augmented ledger* in the extended version of [11]). Fixing the witnesses in the owned UTXOs "binds" the tags to UTXOs and simplifies reasoning about spends. Indeed, to say that a transaction T spends a UTXO u is to say the *tag* s of u is revealed in T (recall Section 2.2), but the network may only be convinced of the link between s and u via a zero-knowledge *argument*, the soundness of which need not hold against unbounded provers. With witnesses fixed in the owned UTXO $\bar{u}$ corresponding to u, we may simply rely in our model on a deterministic algorithm (predicate) Spends that decides whether a transaction T spends a UTXO u by recomputing the tag of u and comparing with the tag revealed in T.

The relevant functionality of a cryptocurrency can then be reduced to three algorithms. Setup, on input security parameter in unary, returns public parameters pp. $\mathsf{Coinbase}$ produces a state of the ledger L (*e.g.*, the *note commitment tree* in Zcash [40, Section 3.8]) together with a UTXO set U such that all UTXOs $u \in U$ use the same amount (not strictly necessary). Finally, Spends takes a transaction T together with an owned UTXO $\bar{u}$ and returns 1 if T is consistent with the state of the ledger L and spends u; otherwise, it returns 0. Notably, $\mathsf{Coinbase}$ produces the UTXOs honestly according to the *specification* of a cryptocurrency which prescribes more than is normally verified by its *consensus protocol*. For example, where the specification requires a random string be used, a malicious user may use the output of a pseudorandom function instead. The network may have no way of telling. This is elaborated below. Compare also the definition of $\mathsf{Coinbase}$ with the Mint algorithm in [11] or [43]. In general, in prevailing designs, transaction outputs can be sampled independently of the inputs, with zero-knowledge arguments eventually linking the two (see Appendix B). It is therefore fine for us to consider outputs alone in the experiment, detached from the transactions that would have minted them in the real world. Finally, the viewing key disclosure policy, *i.e.*, what the dictator demands to see, is encoded as a set of functions V. Each $v \in V$ maps an owned UTXO $\bar{u}$ to the viewing key $v(\bar{u})$ associated with the owner of u.

Definition 1 considers the following experiment. First, n UTXOs (of equal value and addressed to random recipients) $u_1, \ldots, u_n$ are minted honestly, and the dictator (adversary) is asked to select two *challenge UTXOs* u_{i_0} and u_{i_1}. They are then spent by the spender algorithm $\mathcal{S}$ in two separate transactions T_0 and T_1. Notably, $\mathcal{S}$ may do it in any way it wants to as long as the transactions comply with the consensus protocol (the bit $g = g_0 \wedge g_1$ encodes $\mathcal{S}$'s compliance). The dictator is asked to tell which of u_{i_0} and u_{i_1} was spent in which transaction given the viewing keys for *all* outstanding UTXOs, including u_{i_0} and u_{i_1}. A cryptocurrency has *indistinguishable* spends relative to viewing keys V if no efficient dictator, given access to the viewing keys described by V for every UTXO, can win the above game with non-negligible advantage over guessing. In

that case, the dictator in particular cannot tell whether some UTXO of a given subject is being spent in a given transaction or not. Such *spend attribution* is, however, a minimal requirement for any auditing functionality, as the dictator must be able to first identify any non-compliance in order to prosecute it (*cf.* [22, Insight 9]).

Definition 1. *A cryptocurrency* $\Pi = (\mathsf{Setup}, \mathsf{Coinbase}, \mathsf{Spends})$ has indistinguishable spends relative to viewing keys V *if there exists an efficient (probabilistic polynomial-time) spender algorithm* $\mathcal{S}$ *such that, for all efficient dictators* $\mathcal{D}$ *and all anonymity set sizes* $2 \leq n \leq \mathsf{poly}(\lambda)$*, there exists a negligible function* ε *such that, for any* $b \in \{0,1\}$*,*

$$\Pr[\mathsf{SpendInd}^{b,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}(\lambda) = (\cdot,1)] \geq 1 - \varepsilon(\lambda)\,,$$

and

$$\Big|\Pr[\mathsf{SpendInd}^{0,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}(\lambda) = (1,\cdot)] - \Pr[\mathsf{SpendInd}^{1,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}(\lambda) = (1,\cdot)]\Big| \leq \varepsilon(\lambda)\,,$$

where the experiment $\mathsf{SpendInd}^{b,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}$ *is presented in Figure 1, and the probabilities are taken over the random coins of all algorithms involved.*

$\mathsf{SpendInd}^{b,n,V}_{\Pi,\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)$

$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
$L, \bar{u}_1, \ldots, \bar{u}_n \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pp}, n)$
$\mathbf{v} \leftarrow \big\{\big(v(\bar{u}_1), \ldots, v(\bar{u}_n)\big) \mid v \in V\big\}$
$i_0, i_1, \mathsf{st} \leftarrow\!\!\$\ \mathcal{D}(\mathsf{pp}, L, \mathbf{v}, u_1, \ldots, u_n)$
if $\neg(1 \leq i_0 < i_1 \leq n)$ **then**
  return $(0,1)$
endif
$T_0 \leftarrow\!\!\$\ \mathcal{S}(\mathsf{pp}, L, i_0, i_1, \bar{u}_{i_0}, u_1, \ldots, u_n)$
$g_0 \leftarrow \big(\mathsf{Spends}(\mathsf{pp}, L, T_0, \bar{u}_{i_0}) = 1 \wedge \mathsf{Spends}(\mathsf{pp}, L, T_0, \bar{u}_{i_1}) = 0\big)$
$T_1 \leftarrow\!\!\$\ \mathcal{S}(\mathsf{pp}, L, i_1, i_0, \bar{u}_{i_1}, u_1, \ldots, u_n)$
$g_1 \leftarrow \big(\mathsf{Spends}(\mathsf{pp}, L, T_1, \bar{u}_{i_0}) = 0 \wedge \mathsf{Spends}(\mathsf{pp}, L, T_1, \bar{u}_{i_1}) = 1\big)$
$b' \leftarrow\!\!\$\ \mathcal{D}(\mathsf{st}, T_b, T_{1-b})$
return $(b', g_0 \wedge g_1)$

Fig. 1. Spend indistinguishability experiment

Some comments about the model are in order. Note the quantifiers first: we require that there be *at least one* way for the subjects to transact ("there exists

a spender algorithm $\mathcal{S}$") such that, no matter what the dictator does ("for all dictators $\mathcal{D}$"), they cannot tell apart their subjects' spends. Second, to accurately model the capabilities of subjects, who only know the keys to their own UTXOs, we deliberately call $\mathcal{S}$ twice separately to produce each of T_0 and T_1. Note also at this point that we give the dictator *both* T_0 and T_1 (produced relative to the same ledger state L). Third, note that Definition 1 assumes the UTXOs $u_1, \ldots, u_n$ are minted honestly via $\mathsf{Coinbase}$. Letting the dictator $\mathcal{D}$ mint them instead, *i.e.*, setting $\mathcal{A} = \mathcal{D}$ (which would be more in line with the requirements of [41] and the ideas from [11] to some extent), would unnecessarily complicate the model, and it is not relevant for our purposes. Indeed, if being the sender of a transaction that mints u gives the dictator any advantage, the owner of u suspicious of this could "recycle" u by sending it back to themselves (*e.g.*, as part of making change in a legal transaction, *i.e.*, one cleared by the dictator).[6] Conversely, we could allow $\mathcal{S}$ to produce the UTXOs, thus capturing more sophisticated users' sending the witnesses for u_{i_0} and u_{i_1} completely out of band and only encrypting dummy data on chain. Such a model is considered in Section 4. Finally, one could also fix $\mathcal{S}$ to be the canonical transaction creation algorithm complying fully with the specification (which is almost what the proofs in Appendix D do anyway). The results would then correspond to honest users' transacting with some default (reference) wallet implementations after their viewing keys have been compromised (*e.g.*, leaked by the server with which they were entrusted for the purpose of scanning the ledger [73]).

3.2 Spend Indistinguishability Today

With viewing keys often disclosed to semi-trusted servers, who may then fall prey to cyberattacks, it is crucial to evaluate the impact of any breaches on the privacy of users. Clearly, *full viewing keys* defined by Zcash and Firo (recall Section 2.2) do not afford spend indistinguishability. Indeed, their very purpose is the computation of tags, which are unambiguously associated to notes in order to prevent double-spending. (This is an example of the concordance between accountability and the consensus protocol observed in [22,25].) More commonly, however, only incoming viewing keys are disclosed. This then warrants the evaluation of spend indistinguishability relative to them.

Theorem 1. *Zcash has indistinguishable spends in the random oracle model relative to the incoming and outgoing viewing keys* $\{\mathsf{ivk}, \mathsf{ovk}\}$.

The above theorem roughly corresponds to the security requirement of [40, Section 4.16]. Its proof in Appendix D.1 seems to be the first one that is publicly available, but the reference implementation's documentation asserts spend indistinguishability ("unlinkability" therein) based on the pseudorandomness of the tag derivation function.[7] An analogous result holds for Firo and is proved

[6] Recall that output notes are minted using independent coin tosses and only linked to transaction inputs via tags and zero-knowledge arguments.

[7] See `https://zcash.github.io/orchard/design/nullifiers.html` (accessed: 30 August 2026).

in Appendix D.2. Notably, the statement is strictly stronger than the natural corollary of ledger indistinguishability (LIND), whose proof is sketched in [43], as the dictator in the present work gets to see the incoming viewing keys of all outstanding UTXOs.

Theorem 2. *Firo has indistinguishable spends in the random oracle model relative to the incoming viewing key* $\mathsf{addr}_{\mathrm{in}}$ *under the Decisional Diffie-Hellman Inverse (DDHI) assumption.*

For Monero, on the other hand, spend indistinguishability relative to the sole viewing key k^v seems to be "folklore" and taken for granted by the cryptocurrency community with little to no formal justification. It is implicitly assumed to follow from *chosen-target linkable anonymity* of CLSAG ring signatures (see Definition 3 in Appendix C). It is, however, not clear how to, given only CLSAG public keys $\{(K_i^o, Z_i)\}_{i=1}^n$, simulate, in the reduction, the transactions T_0 and T_1 for the dictator $\mathcal{D}$. This is because of their directly observable algebraic structure. In Zcash and Firo, inputs and outputs of a transaction are shown to balance with a zero-knowledge argument, but, in Monero, a direct algebraic relation between the value commitments is checked by the validators which cannot be simulated in a polynomial-time reduction. Specifically, to simulate even just one of the transactions, say, T_0, the reduction would need to sample (at least) three Pedersen commitments: C_{i_0}, C_{i_1}, and C' that satisfy

$$C' = C_{i_0} - Z_{i_0} = C_{i_1} - Z_{i_1}\,, \tag{1}$$

and such that trapdoors of C_{i_0} and C_{i_1} are known, as they get encrypted under k^v in u_{i_0} and u_{i_1}, respectively (see Appendix B.1). To do this, the reduction would have to break the discrete logarithm problem. The same argument applies to any underlying ring signature (including the previously used MLSAG [46]). Indeed, it is easily seen that no black-box reduction to any anonymity property of the ring signature is possible as long as the signing keys are derived from the openings of the value commitments, which must be disclosed to $\mathcal{D}$. The implicit belief of the Monero community (as well as a silent assumption in [25]) is thus merely reformulated below in the language of Definition 1.

Conjecture 1. Monero has indistinguishable spends relative to the incoming viewing key k^v.

While the above argument precludes a black-box reduction from *any* anonymity notion of the underlying ring signature scheme, we remark that, in particular, CLSAG's anonymity in Definition 3, quoted from [39], is formalized wrong, and, as written, it is unsatisfiable by *any* linkable ring signature scheme. We elaborate on the issue in Appendix C where we also provide a fix.

4 General Oversight with Viewing Keys

This next section studies how (in)effective viewing keys are at facilitating financial oversight of uncooperative subjects. As above, consider the practical setting

(minimizing on-chain data), where subjects are supposed to submit transactions to an off-chain audit under legal obligation alone [10, 25] (or to disassociate themselves from criminals [18]). Before, it was appropriate for us to limit our attention to honestly minted UTXOs. Indeed, the model of Definition 1 applies to a "crypto-asset service provider" [70] undergoing an audit or a lightweight wallet offloading responsibility to an external server. It does not capture, however, the general setting of oversight built around viewing keys. We must instead consider a more elaborate experiment, where notes may be formatted maliciously. In particular, their corresponding witnesses may be sent completely out of band.

4.1 Spend Indistinguishability Under Off-Chain Communication

We capture the above setting by considering an additional spender algorithm $\mathcal{S}'$ that produces the UTXOs $u_1, \ldots, u_n$ instead of $\mathsf{Coinbase}$. The output of $\mathcal{S}'$ must naturally be indistinguishable from that of $\mathsf{Coinbase}$ from the point of view of parties without any viewing keys. The experiment then proceeds as before. $\mathcal{S}$ receives the owned UTXOs $\bar{u}_{i_0}$ and $\bar{u}_{i_1}$, but now u'_{i_0} and u'_{i_1} are not guaranteed to contain witnesses encrypted under any viewing key in V. They may instead be "shadow UTXOs" whose witnesses are sent off-chain or hidden in the UTXOs in some non-standard way.

Definition 2. *A cryptocurrency* $\Pi = (\mathsf{Setup}, \mathsf{Coinbase}, \mathsf{Spends})$ has indistinguishable spends relative to viewing keys V in the presence of out-of-band channels *if there exist efficient spender algorithms* $\mathcal{S}$ *and* $\mathcal{S}'$ *such that, for all efficient dictators* $\mathcal{D}$ *and all anonymity set sizes* $2 \leq n \leq \mathsf{poly}(\lambda)$*, there exists a negligible function* ε *such that*

$$\left| \Pr\left[\mathcal{D}(\mathsf{pp}, L, u_1, \ldots, u_n) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\; \mathsf{Setup}(1^\lambda) \\ L, \bar{u}_1, \ldots, \bar{u}_n \leftarrow\!\$\; \mathsf{Coinbase}(\mathsf{pp}, n) \end{array} \right] \right.$$
$$\left. - \Pr\left[\mathcal{D}(\mathsf{pp}, L, u_1, \ldots, u_n) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\$\; \mathsf{Setup}(1^\lambda) \\ L, \bar{u}_1, \ldots, \bar{u}_n \leftarrow\!\$\; \mathcal{S}'(\mathsf{pp}, n) \end{array} \right] \right| \leq \varepsilon(\lambda)\,,$$

we have, for any $b \in \{0, 1\}$*,*

$$\Pr[\mathsf{SpendInd}^{b,n,V}_{\Pi,\mathcal{S}',\mathcal{S},\mathcal{D}}(\lambda) = (\cdot, 1)] \geq 1 - \varepsilon(\lambda)\,,$$

and

$$\left| \Pr[\mathsf{SpendInd}^{0,n,V}_{\Pi,\mathcal{S}',\mathcal{S},\mathcal{D}}(\lambda) = (1, \cdot)] - \Pr[\mathsf{SpendInd}^{1,n,V}_{\Pi,\mathcal{S}',\mathcal{S},\mathcal{D}}(\lambda) = (1, \cdot)] \right| \leq \varepsilon(\lambda)\,,$$

where the experiment $\mathsf{SpendInd}^{b,n,V}_{\Pi,\mathcal{S}',\mathcal{S},\mathcal{D}}$ *is presented in Figure 1, and the probabilities are taken over the random coins of all algorithms involved.*

As a technicality, observe that we do not prevent $\mathcal{S}'$ from minting two notes with identical tags (tag non-collision is only verified by the network upon spending and not at the time of minting). Explicitly banning it is redundant, however, as a tag collision between $\bar{u}_{i_0}$ and $\bar{u}_{i_1}$ fails the compliance check and forces $g_0 = g_1 = 0$.

Theorem 3. *Zcash (resp. Firo) has indistinguishable spends in the random oracle model (resp. in the ROM and under the DDHI assumption) relative to the full viewing key* fvk *(resp.* $\mathsf{addr}_{\mathrm{full}}$*) in the presence of out-of-band channels.*

Proof (sketch). Let $\mathcal{S}'$ produce all ciphertexts in $u_1, \ldots, u_n$ from uniformly random keys and plaintexts. Since their consistency is not verified by the consensus protocol, the output of $\mathcal{S}'$ is publicly consistent with the output of $\mathsf{Coinbase}$. $\mathcal{S}$ can still spend u_{i_0} and u_{i_1} because it is given the corresponding witnesses out of band. The remaining note components are then perfectly (resp. statistically) indistinguishable from uniformly random. Transactions T_0 and T_1 then similarly do not give the dictator with fvk (resp. $\mathsf{addr}_{\mathrm{full}}$) any (resp. any non-negligible amount of) information.[8] □

Theorem 4. *Monero has indistinguishable spends in the random oracle model relative to the incoming viewing key* k^v *in the presence of out-of-band channels under the assumption that CLSAG is linkably anonymous.*

Proof (sketch). Let $\mathcal{S}'$ fix $C' = \delta \cdot G + v \cdot H$ and mint each UTXO $u_i = (K_i^o, R_i, C_i, c_i)$ with value commitment $C_i \leftarrow C' + z_i \cdot G$, R_i and c_i uniform and independent of $K_i^o = K_i^s + \mathsf{Hash}_{\mathbb{F}_q}(r_i' \cdot K_i^v) \cdot G$. The Diffie-Hellman share $R_i' = r_i' \cdot G$ is included in the owned UTXO $\bar{u}_i$. The spender $\mathcal{S}$ can then create any transaction T with pseudo-output commitment C' and two outputs $u^{(1)}$ and $u^{(2)}$ with value commitments, respectively,

$$C^{(1)} = y \cdot G + v \cdot H \quad \text{and} \quad C^{(2)} = (\delta - y) \cdot G + 0 \cdot H\,,$$

for uniform y. The commitments $C^{(1)}$ and $C^{(2)}$ are individually uniform but sum to C' as required by the consensus protocol (see Appendix B.1). The spender $\mathcal{S}$ also knows all the required openings to produce the range proofs π and the CLSAG signatures σ. To show that the pair $(\mathcal{S}', \mathcal{S})$ is a witness for the theorem, reduce from chosen-target linkable anonymity of CLSAG (Definition 3). Sample uniform C' and, for each CLSAG public key (K_i^o, Z_i), simulate the corresponding "shadow UTXO" u_i by setting $u_i \leftarrow (K_i^o, R_i, C' + Z_i, c_i)$, for R_i and c_i uniform. Each transaction

$$T_b = \big(\{u_1, \ldots, u_n\}, C', \sigma_b, \tilde{K}_b, \tilde{Z}_b, \pi, \{u^{(1)}, u^{(2)}\}\big)$$

can then be simulated by first simulating the Bulletproofs+ [24] range proof π_b, and obtaining the signature σ_b together with $\tilde{K}_b$ and $\tilde{Z}_b$ from the CLSAG oracle (see Figure 5). The simulation is perfect, and whatever advantage $\mathcal{D}$ has in the spend indistinguishability game (Figure 1) is also the advantage of the reduction $\mathcal{A}$ in $\mathsf{LinkAnon}^n_{\mathcal{A},\Sigma}$ (Figure 5). □

While Theorem 4 holds, the proof is in a sense unsatisfactory. Indeed, note how the reduction requires that $\mathcal{S}$ reuse the same pseudo-output commitment C'

[8] The computational loss in Firo results from the tag N being computed via the verifiable pseudorandom function of Dodis and Yampolskiy [31] and not via a random oracle as in the case of Zcash (see Equation (5)).

across both transactions T_0 and T_1 to be able to simulate $\mathcal{D}$'s view given only the public keys Z_i. Such a collision would otherwise (for canonical spenders) happen only with negligible probability. This is again the public algebraic structure of Monero transactions causing problems. Whether a more natural pair $(\mathcal{S}', \mathcal{S})$ exists that witnesses Definition 2 remains an open question.

Anamorphic Spending. Note that, in the real world, no actual out-of-band communication needs to take place. Indeed, the witnesses could still be placed on chain but encrypted under an independent *double key* dkey whose existence a subject can plausibly deny [48, 49, 59]. This "hidden in plain sight" approach reflects a real concern of criminals which is minimizing interactions with their accomplices that could leave traces for the law enforcement to follow [25, 44]. Here, they seem to only ever be interacting with the blockchain, no extra out-of-band traffic. Furthermore, ideas from *anamorphic spending*, introduced in [25], could be used, whereby the subject, call her Alice, sends any witnesses to her accomplice, Bob, using subliminal channels afforded by the cryptocurrency itself [4, 13, 25]. For Monero, naturally, the biggest concern in this regard remains the vast covert bandwidth available in CLSAG signatures [39] recognized in [25] (see Appendix C). Surprisingly, however, what becomes a concern for Zcash and Firo is the "uncontrolled randomness" [4] in state-of-the-art zero-knowledge arguments. These may accommodate wide subliminal channels through the standard technique of *rejection sampling* alone [59]. For example, a typical 2-input-2-output transaction involves, due to proofs alone (*i.e.*, discounting signatures, value commitments, randomized public keys, *etc.*), 17 independently rerandomizable 256-bit elliptic curve points in Monero, 32 in Firo, and 77 in Zcash! Each such point P can be rejection-sampled until the evaluations $\mathsf{F}_{\mathsf{dkey}}(P)$ spell out "chunks" of some covert message m for a pseudorandom function $\mathsf{F}_{\mathsf{dkey}}$ keyed with the double key dkey shared between Alice and Bob. This m can contain any missing witnesses for notes already minted on chain or even entire transactions appropriately compressed to be broadcast by a proxy (*cf.* the no-proxy assumption underlying deanonymizations today [30]). In general, an entire underground payment network can be built that settles on the same public ledger, right under the nose of the dictator lulled into a false sense of security by mandating the disclosure of the full viewing key.

4.2 Conclusions and Remedy

What the previous section showed is that existing viewing keys are ineffective as a general auditability instrument when *all* monitored parties are malicious. Specifically, if a dictator were to demand the disclosure of all viewing keys for Zcash and Firo today, their subjects could first "conceal" their coins (*e.g.*, during *vacatio legis*) by sending them back to themselves, this time formatting the notes like $\mathcal{S}'$ above would, namely, not encrypting any witnesses under any compromised viewing keys. The dictator would then have no way of recognizing the new "shadow UTXOs" even if the subjects truthfully surrendered their viewing keys.

Fundamentally, this is because the consensus protocol is not concerned in any way with verifying the consistency of the ciphertexts, whereas viewing key-based oversight currently critically relies on these ciphertexts being consistent.

Let us now briefly sketch a lightweight change to existing privacy coin protocols that would remedy this while minimizing on-chain data (*cf.* [35, 68]). In doing so, we admittedly depart from the anamorphic cryptography setting, where we assume no changes to the underlying protocols are possible. It is, however, instructive to see how auditability can be facilitated with minor changes to consensus rules. Let a be a subject's address and v the viewing key associated to a that has been surrendered to the dictator. The dictator wants an "oracle" that, given a UTXO u and the key v, verifies if u is addressed to a without any online interaction with any subject. Note that, once ownership has been established this way, then the dictator can obligate the subject to attest to additional details of u (compare with the designs in [10,18,25]). To this end, suppose we extend every UTXO u to include an *attribution tag* A that only enables the holder of v to *link* the note u with the recipient's address a. Specifically, instead of publishing u in the ledger, publish $\hat{u} = (u, A)$, where $A = \big(r, R_v, \pi\big)$, for uniformly random r, $R_v = \mathsf{F}_v(r)$ for a pseudorandom function F keyed with the viewing key v, and a zero-knowledge argument π showing that R_v was computed correctly as $\mathsf{F}_v(r)$ for v which corresponds to the address a of the recipient of u. Pseudorandomness of F and zero knowledge of π then guarantee that privacy properties of the original cryptocurrency remain intact. At the same time, the dictator can use their knowledge of v to efficiently and, most importantly, in an offline fashion identify UTXOs in the possession of their subjects.

Three points are important to make. First, validity of π must be checked by the consensus protocol. This is the "anchor" that convinces the dictator that, if $R_v \neq \mathsf{F}_{v'}(r)$, then (except with negligible probability) $v \neq v'$, and so the note is not owned by the subject corresponding to v'. Second, the pseudorandom function F_v needs to have some algebraic structure in order to be evaluated by the sender who mints u. This is in general *not* the subject corresponding to the viewing key v (who is instead the recipient of u). In the case of Monero, for example, we could have $\mathsf{F}_{k^v}(R) := k^v \cdot R$, which is a pseudorandom function under the Decisional Diffie-Hellman (DDH) assumption. Finally, note that, even if we extended UTXOs with attribution tags, the dictator would still not be able to take for granted the consistency of the ciphertexts published on chain and would separately need to demand proofs about the structure of transactions [10,18]. In particular, the dictator should have the relevant subject reveal the tag s of the (now correctly attributed) UTXO u and prove its consistency (*cf.* [25]). With this, the dictator would be able to recognize when this asset gets spent and request appropriate attestations and disclosure at that point also. The auditing functionality would then finally be sound, in that no transaction would go unaccounted for: inbound transfers would be identified by their attribution tags A and outbound transfers by their (regular) tags s. Refusal to submit a transaction to an audit could be prosecuted accordingly (*cf.* [22, Insight 9]). This is of particular importance when dealing with institutional exposure to privacy coins,

which, *e.g.*, the AML regulation in Europe currently aims to strictly prohibit (see [71, Article 79(1)]).

In closing, observe that, in many practical settings, the stronger model of Definition 1 may in fact be more appropriate. For example, during a withdrawal to a self-hosted wallet (which is paid extra attention by the regulators; see, *e.g.*, [67] or [69, Article 14(5)]), it is reasonable to assume that a licensed exchange would mint the UTXOs for the wallet honestly. The simpler security model of Definition 1 then accurately describes monitoring the client afterwards.

5 Summary and Closing Remarks

Despite the European Court of Human Rights' establishing in a landmark ruling [34] that

> *[permitting] the public authorities to have access, on a generalised basis and without sufficient safeguards, to the content of electronic communications . . . impairs the very essence of the right to respect for private life[,]*

it seems no such principle applies to privacy in one's economic life, and the legal crackdown on anonymity-enhancing cryptocurrencies intensifies. Indeed, cryptocurrencies have become a new front of the so-called "crypto wars" [1,2], with legislators willing to sacrifice financial privacy on the altar of accountability and transparency. To not risk their banning privacy coins for good in the conviction they are nothing more than laundromats [5], it becomes crucial to identify solid mechanisms for oversight that strike a balance between lawful access and data minimization [68], particularly in the age of AI-powered financial surveillance. Viewing keys are often proposed as a means of addressing this issue [26,27,33,58], but the security around them so far lacked systematic evaluation.

The present paper has laid the groundwork for evaluating both existing solutions, namely, Zcash, Monero, and Firo, as well as future designs, such as, *e.g.*, the upcoming Monero releases: Seraphis and Jamtis, by introducing two models of spend indistinguishability relative to viewing keys (Definitions 1 and 2). In the latter model, we showed that *all* viewing keys defined today for Zcash and Firo are ineffective against sophisticated criminals determined to evade AML monitoring. Indeed, they may happily comply with any demand to surrender even the most powerful viewing keys and nonetheless continue to run covert operations on the public ledger itself (Theorem 3). Notably, however, while not a bulletproof solution, viewing keys have their place in a risk-based approach to combating, say, money laundering or financing of terrorism through institutional actors such as exchanges (*cf.* the exemption of person-to-person transfers in [69, Article 2(4)]), where the stronger model of Definition 1 may apply instead. We also filled in the gaps in provable anonymity of Zcash and Firo in a more moderate setting, where only the incoming viewing keys are compromised (Theorems 1 and 2). Finally, we identified a flaw in the design of Monero, namely, the publicly observable algebraic structure of transactions, which hinders provably establishing spend indistinguishability relative to its sole viewing

key despite this being assumed already by its users. Resolving Conjecture 1 and generalizing Theorem 4 are thus urgent next steps. They entail one of the following: changes to the Monero protocol, reformulating CLSAG [39] security in the language of zero knowledge and simulatability (compare with [43]), or doing a white-box reduction from the DDH problem directly. This is particularly important given that incoming viewing keys find everyday use in offloading blockchain scanning to third parties [74].

References

1. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Callas, J., Diffie, W., Landau, S., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., Teague, V., Troncoso, C.: Bugs in our pockets: the risks of client-side scanning. Journal of Cybersecurity **10**(1), tyad020 (01 2024). https://doi.org/10.1093/cybsec/tyad020, https://doi.org/10.1093/cybsec/tyad020
2. Abelson, H., Anderson, R., Bellovin, S.M., Benaloh, J., Blaze, M., Diffie, W., Gilmore, J., Green, M., Landau, S., Neumann, P.G., Rivest, R.L., Schiller, J.I., Schneier, B., Specter, M.A., Weitzner, D.J.: Keys under doormats: mandating insecurity by requiring government access to all data and communications ‡. Journal of Cybersecurity **1**(1), 69–79 (11 2015). https://doi.org/10.1093/cybsec/tyv009, https://doi.org/10.1093/cybsec/tyv009
3. Abramova, S., Schöttle, P., Böhme, R.: Mixing coins of different quality: A game-theoretic approach. In: Brenner, M., Rohloff, K., Bonneau, J., Miller, A., Ryan, P.Y., Teague, V., Bracciali, A., Sala, M., Pintore, F., Jakobsson, M. (eds.) Financial Cryptography and Data Security. pp. 280–297. Springer International Publishing, Cham (2017)
4. Alsalami, N., Zhang, B.: Uncontrolled randomness in blockchains: Covert bulletin board for illicit activity. In: 28th IEEE/ACM International Symposium on Quality of Service, IWQoS 2020, Hangzhou, China, June 15-17, 2020. pp. 1–10. IEEE (2020). https://doi.org/10.1109/IWQOS49365.2020.9213064, https://doi.org/10.1109/IWQoS49365.2020.9213064
5. Anderson, R., Shumailov, I., Ahmed, M.: Making Bitcoin legal. In: Matyáš, V., Švenda, P., Stajano, F., Christianson, B., Anderson, J. (eds.) Security Protocols XXVI. pp. 243–253. Springer International Publishing, Cham (2018)
6. Anderson, R., Shumailov, I., Ahmed, M., Rietmann, A.: Bitcoin redux. In: Workshop on the Economics of Information Security (WEIS). Innsbruck, Austria (2018)
7. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in Bitcoin. In: Sadeghi, A.R. (ed.) Financial Cryptography and Data Security. pp. 34–51. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
8. Baldimtsi, F., Chalkias, K.K., Madathil, V., Roy, A.: SoK: Privacy-preserving transactions in blockchains. Cryptology ePrint Archive, Paper 2024/1959 (2024), https://eprint.iacr.org/2024/1959
9. Balthazor, A.: The bona fide acquisition rule applied to cryptocurrency. Georgetown Law Technology Review **3**(2) (2019), https://georgetownlawtechreview.org/the-bona-fide-acquisition-rule-applied-to-cryptocurrency/GLTR-05-2019
10. Beal, J., Fisch, B.: Derecho: Privacy pools with proof-carrying disclosures. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. p. 3197–3211. CCS '24, Association for Computing Machinery,

New York, NY, USA (2024). https://doi.org/10.1145/3658644.3670270, https://doi.org/10.1145/3658644.3670270
11. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474 (2014). https://doi.org/10.1109/SP.2014.36
12. Biryukov, A., Feher, D.: Privacy and linkability of mining in Zcash. In: 2019 IEEE Conference on Communications and Network Security (CNS). pp. 118–123 (2019). https://doi.org/10.1109/CNS.2019.8802711
13. Biryukov, A., Feher, D., Vitto, G.: Privacy aspects and subliminal channels in Zcash. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. p. 1813–1830. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3319535.3345663, https://doi.org/10.1145/3319535.3345663
14. Biryukov, A., Khovratovich, D., Tikhomirov, A.: Privacy-preserving KYC on Ethereum. In: Proceedings of the ERCIM-Blockchain 2018 Workshop on Blockchain Engineering: Challenges and Opportunities for Computer Science Research. European Research Consortium for Informatics and Mathematics (2018)
15. BITCRIME Project: Prävention von Straftaten mit Bitcoins und Alt-Coins (2017), https://www.bitcrime.de/presse-publikationen/pdf/BITCRIME-RegulRep.pdf, accessed: 26 November 2025
16. Bowe, S., Grigg, J., Hopwood, D.: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Paper 2019/1021 (2019), https://eprint.iacr.org/2019/1021
17. Bünz, B., Agrawal, S., Zamani, M., Boneh, D.: Zether: Towards privacy in a smart contract world. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security. pp. 423–443. Springer International Publishing, Cham (2020)
18. Buterin, V., Illum, J., Nadler, M., Schär, F., Soleimani, A.: Blockchain privacy and regulatory compliance: Towards a practical equilibrium. Blockchain: Research and Applications **5**(1), 100176 (2024). https://doi.org/https://doi.org/10.1016/j.bcra.2023.100176, https://www.sciencedirect.com/science/article/pii/S2096720923000519
19. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005. pp. 302–321. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
20. Chainalysis: The 2024 crypto crime report (2024), https://go.chainalysis.com/crypto-crime-2024.html
21. Chainalysis: The 2025 crypto crime report (2025), https://go.chainalysis.com/2025-Crypto-Crime-Report.html
22. Chatzigiannis, P., Baldimtsi, F., Chalkias, K.: SoK: Auditability and accountability in distributed payment systems. In: International Conference on Applied Cryptography and Network Security (ACNS 2021). pp. 311–337. Springer (2021). https://doi.org/10.1007/978-3-030-78375-4_13
23. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology. pp. 199–203. Springer US, Boston, MA (1983)
24. Chung, H., Han, K., Ju, C., Kim, M., Seo, J.H.: Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. IEEE Access **10**, 42081–42096 (2022). https://doi.org/10.1109/ACCESS.2022.3167806

25. Cinal, A., Kubiak, P., Kutyłowski, M., Wechta, G.: Anamorphic Monero transactions: The threat of bypassing anti-money laundering laws. In: Nicomette, V., Benzekri, A., Boulahia-Cuppens, N., Vaidya, J. (eds.) Computer Security – ESORICS 2025. pp. 103–123. Springer Nature Switzerland, Cham (2026)
26. ComplyFirst: Compliance brief: Monero (XMR) (2020), https://s33483.pcdn.co/wp-content/uploads/2020/12/ComplyFirst-Compliance-Brief-for-Monero-XMR.pdf, accessed: 24 August 2026
27. ComplyFirst: Compliance brief: Zcash (ZEC) (2020), https://s33483.pcdn.co/wp-content/uploads/2020/12/ComplyFirst-Compliance-Brief-for-Zcash-ZEC.pdf, accessed: 24 August 2026
28. Damgård, I., Ganesh, C., Khoshakhlagh, H., Orlandi, C., Siniscalchi, L.: Balancing privacy and accountability in blockchain identity management. In: Paterson, K.G. (ed.) Topics in Cryptology – CT-RSA 2021. pp. 552–576. Springer International Publishing, Cham (2021)
29. De Filippi, P.: The interplay between decentralization and privacy: the case of blockchain technologies. Journal of Peer Production (7) (Sep 2016), https://papers.ssrn.com/abstract=2852689
30. Deuber, D., Ronge, V., Rückert, C.: SoK: Assumptions underlying cryptocurrency deanonymizations. Proceedings on Privacy Enhancing Technologies **2022**(3), 670–691 (2022). https://doi.org/10.56553/popets-2022-0091
31. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) Public Key Cryptography - PKC 2005. pp. 416–431. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
32. Egger, C., Lai, R.W.F., Ronge, V., Woo, I.K.Y., Yin, H.H.F.: On defeating graph analysis of anonymous transactions. Proceedings on Privacy Enhancing Technologies (PoPETs) **2022**(3), 538–557 (2022). https://doi.org/10.56553/popets-2022-0085
33. Electric Coin Company: Zcash regulatory & compliance brief (2019), https://bitzecbzc.github.io/wp-content/uploads/2019/04/Zcash-Regulatory-Brief.pdf, accessed: 24 August 2026
34. European Court of Human Rights: Case of Podchasov v. Russia (2024), https://hudoc.echr.coe.int/fre?i=001-230854, accessed: 26 October 2025
35. European Data Protection Board: Guidelines 02/2025 on processing of personal data through blockchain technologies (2026), https://www.edpb.europa.eu/system/files/2026-07/edpb_guidelines_202502_blockchain_v2_en.pdf, accessed: 20 September 2026
36. Europol: IOCTA, internet organised crime threat assessment 2024 (2024). https://doi.org/10.2813/442713
37. Fauzi, P., Meiklejohn, S., Mercer, R., Orlandi, C.: Quisquis: A new design for anonymous cryptocurrencies. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 649–678. Springer International Publishing, Cham (2019)
38. Garman, C., Green, M., Miers, I.: Accountable privacy for decentralized anonymous payments. In: Financial Cryptography and Data Security. pp. 81–98. Springer (2016)
39. Goodell, B., Noether, S., Blue, A.: Concise linkable ring signatures and forgery against adversarial keys. eprint.iacr.org **654** (2019), https://eprint.iacr.org/2019/654
40. Hopwood, D.E., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification (2026), https://zips.z.cash/protocol/protocol.pdf, accessed: 6 August 2026

41. Hopwood, D.E., Grigg, J.: Security properties of Sapling viewing keys. Zcash Improvement Proposal 310. Created March 09, 2020. (2020), https://zips.z.cash/zip-0310, accessed: 26 November 2025
42. Ibikunle, G., Mollica, V., Sun, Q.: Why so many coins? examining the demand for privacy-preserving cryptocurrencies. The British Accounting Review p. 101637 (2025). https://doi.org/10.1016/j.bar.2025.101637, https://www.sciencedirect.com/science/article/pii/S0890838925000873
43. Jivanyan, A., Feickert, A.: Lelantus Spark: Secure and flexible private transactions. In: Matsuo, S., Gudgeon, L., Klages-Mundt, A., Perez Hernandez, D., Werner, S., Haines, T., Essex, A., Bracciali, A., Sala, M. (eds.) Financial Cryptography and Data Security. FC 2022 International Workshops. pp. 409–447. Springer International Publishing, Cham (2023)
44. Juels, A., Kosba, A., Shi, E.: The ring of Gyges: Investigating the future of criminal smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 283–295. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978362, https://doi.org/10.1145/2976749.2978362
45. Kappos, G., Yousaf, H., Maller, M., Meiklejohn, S.: An empirical analysis of anonymity in Zcash. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 463–477. USENIX Association, Baltimore, MD (Aug 2018), https://www.usenix.org/conference/usenixsecurity18/presentation/kappos
46. Koe, Alonso, K.M., Noether, S.: Zero to Monero: Second edition (2020), https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf
47. Kumar, A., Fischer, C., Tople, S., Saxena, P.: A traceability analysis of Monero's blockchain. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security – ESORICS 2017. pp. 153–173. Springer International Publishing, Cham (2017)
48. Kutyłowski, M., Persiano, G., Phan, D., Yung, M., Zawada, M.: The self-anticensorship nature of encryption: On the prevalence of anamorphic cryptography. Proceedings on Privacy Enhancing Technologies **2023**, 170–183 (10 2023). https://doi.org/10.56553/popets-2023-0104
49. Kutyłowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: Anamorphic signatures: Secrecy from a dictator who only permits authentication! In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 759–790. Springer Nature Switzerland, Cham (2023)
50. Lai, R.W.F., Ronge, V., Ruffing, T., Schröder, D., Thyagarajan, S.A.K., Wang, J.: Omniring: Scaling private payments without trusted setup. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. p. 31–48. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3319535.3345655, https://doi.org/10.1145/3319535.3345655
51. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference. p. 127–140. IMC '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2504730.2504747, https://doi.org/10.1145/2504730.2504747
52. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from Bitcoin. In: 2013 IEEE Symposium on Security and Privacy. pp. 397–411 (2013). https://doi.org/10.1109/SP.2013.34
53. Möser, M., Böhme, R., Breuker, D.: Towards risk scoring of Bitcoin transactions. In: Financial Cryptography and Data Security. pp. 16–32. Springer (2014)

54. Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Hogan, K., Hennessey, J., Miller, A., Narayanan, A., Christin, N.: An empirical analysis of traceability in the Monero blockchain. Proceedings on Privacy Enhancing Technologies **2018**(3), 143–163 (2018). https://doi.org/10.1515/popets-2018-0025
55. Möser, M., Böhme, R., Breuker, D.: An inquiry into money laundering tools in the Bitcoin ecosystem. In: 2013 APWG eCrime Researchers Summit. pp. 1–14 (2013). https://doi.org/10.1109/eCRS.2013.6805780
56. Möser, M., Narayanan, A.: Effective cryptocurrency regulation through blacklisting (2019), https://maltemoeser.de/paper/blacklisting-regulation.pdf, manuscript
57. Papadoulis, G., Balla, D., Grontas, P., Pagourtzis, A.: AQQUA: Augmenting Quisquis with auditability. In: Fischlin, M., Moonsamy, V. (eds.) Applied Cryptography and Network Security. pp. 283–312. Springer Nature Switzerland, Cham (2025)
58. Perkins Coie LLP: Anti-money laundering regulation of privacy-enabling cryptocurrencies (2020), accessed: 24 August 2026
59. Persiano, G., Phan, D.H., Yung, M.: Anamorphic encryption: Private communication against a dictator. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022. pp. 34–63. Springer International Publishing, Cham (2022)
60. Poelstra, A.: Mimblewimble (2016), https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.pdf
61. PwC: PwC global crypto regulation report 2025 (2025), https://legal.pwc.de/en/services/pwc-legals-eu-regulatory-compliance-operations/pwcs-global-crypto-regulation-report
62. Quesnelle, J.: On the linkability of Zcash transactions (2017)
63. Reid, F., Harrigan, M.: An Analysis of Anonymity in the Bitcoin System, pp. 197–223. Springer New York, New York, NY (2013). https://doi.org/10.1007/978-1-4614-4139-7_10, https://doi.org/10.1007/978-1-4614-4139-7_10
64. van Saberhagen, N.: CryptoNote v2.0 (2013), https://www.getmonero.org/resources/research-lab/pubs/cryptonote-whitepaper.pdf
65. Scharnowski, S.: Dark web traffic, privacy coins, and cryptocurrency trading activity. Finance Research Letters **67**, 105875 (2024). https://doi.org/10.1016/j.frl.2024.105875, https://www.sciencedirect.com/science/article/pii/S154461232400905X
66. Spagnuolo, M., Maggi, F., Zanero, S.: BitIodine: Extracting intelligence from the Bitcoin network. In: Christin, N., Safavi-Naini, R. (eds.) Financial Cryptography and Data Security. pp. 457–468. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
67. Swiss Financial Market Supervisory Authority FINMA: FINMA guidance 02/2019: Payments on the blockchain (2019), https://www.finma.ch/en/news/2019/08/20190826-mm-kryptogwg, accessed: 26 November 2025
68. The European Parliament and the Council of the European Union: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016)
69. The European Parliament and the Council of the European Union: Regulation (EU) 2023/1113 of the European Parliament and of the Council of 31 May 2023 on information accompanying transfers of funds and certain crypto-assets and amending Directive (EU) 2015/849 (2023)

70. The European Parliament and the Council of the European Union: Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on markets in crypto-assets, and amending Regulations (EU) no 1093/2010 and (EU) no 1095/2010 and Directives 2013/36/EU and (EU) 2019/1937 (2023)
71. The European Parliament and the Council of the European Union: Regulation (EU) 2024/1624 of the European Parliament and of the Council of 31 May 2024 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing (2024)
72. Vijayakumaran, S.: Analysis of CryptoNote transaction graphs using the Dulmage-Mendelsohn decomposition. In: Bonneau, J., Weinberg, S.M. (eds.) 5th Conference on Advances in Financial Technologies (AFT 2023). Leibniz International Proceedings in Informatics (LIPIcs), vol. 282, pp. 28:1–28:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2023). https://doi.org/10.4230/LIPIcs.AFT.2023.28, https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.AFT.2023.28
73. Wicht, F.X., Wang, Z., Le, D.V., Cachin, C.: A transaction-level model for blockchain privacy. In: Clark, J., Shi, E. (eds.) Financial Cryptography and Data Security. pp. 293–310. Springer Nature Switzerland, Cham (2025)
74. Wüst, K., Matetic, S., Schneider, M., Miers, I., Kostiainen, K., Čapkun, S.: ZLiTE: Lightweight clients for shielded Zcash transactions using trusted execution. In: Goldberg, I., Moore, T. (eds.) Financial Cryptography and Data Security. pp. 179–198. Springer International Publishing, Cham (2019)
75. Yu, Z., Au, M.H., Yu, J., Yang, R., Xu, Q., Lau, W.F.: New empirical traceability analysis of CryptoNote-style blockchains. In: Goldberg, I., Moore, T. (eds.) Financial Cryptography and Data Security. pp. 133–149. Springer International Publishing, Cham (2019)

A Related Work

Regulation of cryptocurrency trade, as relevant to this work, is given treatment in [5, 6, 9, 53, 56]. Balancing accountability against privacy is the focus of [10, 14, 18, 22, 28, 38, 44], with [38] in particular proposing modifications to Zcash that facilitate oversight. Anonymity guarantees of Bitcoin (or lack thereof) are surveyed in [7, 51, 55, 63, 66], while the privacy of Monero is evaluated in [32, 47, 54, 72, 73, 75] and that of Zcash in [12, 45, 62, 73]. Deuber *et al.* in [30] systematize the (often implicit) assumptions underlying analyses such as those above. Anamorphic cryptography is introduced in [48,49,59], while covert channels in privacy-oriented cryptocurrencies are constructed and studied in [4,13,25].

B Monero, Zcash, and Firo in More Detail

This section gives a more comprehensive introduction to the most relevant privacy coins today, namely, Monero and Zcash, as well as Firo, which is of theoretical interest to the present work. Write $\mathbb{G}$ to denote an elliptic curve group of prime order q (with $|q| = 2\lambda$) over a prime field $\mathbb{F}_p$ and G, H, K, F to denote its generators sampled independently. Write Hash_X to denote a cryptographic hash function with range X.

B.1 Monero

Monero [46], based on CryptoNote [64], is one of the earliest privacy coins to gain mainstream recognition. Its notes have the form:

$$u = (K^o, R, C, c)\,,$$

with $K^o, R, C \in \mathbb{G}$ and c a ciphertext. Users are identified by their public addresses $(K^v, K^s) \in \mathbb{G}^2$. To mint a UTXO u addressed to (K^v, K^s), the originator should sample $r \leftarrow\!\!\$\, \mathbb{F}_q$, set $R \leftarrow r \cdot G$ and

$$K^o \leftarrow K^s + \mathsf{Hash}_{\mathbb{F}_q}(r \cdot K^v) \cdot G\,.$$

We call K^o the (one-time) *stealth address*. The recipient knows the secret spending key k^s and the secret viewing key k^v such that, respectively, $K^s = k^s \cdot G$ and $K^v = k^v \cdot G$. They are therefore able to compute k^o such that $K^o = k^o \cdot G$ by setting

$$k^o \leftarrow k^s + \mathsf{Hash}_{\mathbb{F}_q}(k^v \cdot R)\,.$$

This *output key* is used to authorize the spending of u. C is supposed to be a Pedersen commitment to the amount v attached to u. Specifically, the sender should set

$$C \leftarrow x \cdot G + v \cdot H\,,$$

for some trapdoor x. (Recall that G and H are two generators of $\mathbb{G}$ sampled independently.) The sender should encrypt the witnesses x and v under the Diffie-Hellman key $r \cdot K^v$ and set c to this ciphertext [46]:

$$c \leftarrow\!\!\$\, \mathsf{Enc}_{r \cdot K^v}(x, v)\,.$$

We stress that this is merely prescribed by the specification of Monero, but not enforced by its consensus protocol, hence "should" as opposed to "must." Note how, under normal circumstances, where users adhere to the specification, the viewing key k^v enables the decryption of notes without granting spend authority.

Balance. To prove balance, *i.e.*, that the net output amount equals the net input amount,[9] Monero transactions rely on Pedersen commitments' being homomorphic. Let $C_j = x_j \cdot G + v_j \cdot H$ be the value commitment of the jth input note. For each input note, a new commitment C'_j (referred to as a *pseudo-output commitment* in [46]) is constructed that commits to the same amount v_j but uses a different trapdoor x'_j. The value commitments $\hat{C}_t$ of the outputs (newly minted notes) are then constructed as follows. The trapdoors y_t are selected so that $\sum_j x'_j - \sum_t y_t = 0 \pmod q$. If the input and output amounts balance, then the sum of the pseudo-output commitments will be equal to the sum of the output commitments:

$$\sum_j C'_j - \sum_t \hat{C}_t = \infty\,,$$

[9] Miner's fees are neglected here and in the rest of the paper. They are straightforward to account for.

where ∞ denotes the neutral element of $\mathbb{G}$. This can be publicly checked by the network at large. This design choice, where a direct algebraic relation is verified instead of, say, a zero-knowledge argument has consequences for the provability of the anonymity of Monero (see Conjecture 1). Notably, the above implication only goes one way. Sums of commitments being equal only shows that amounts balance modulo q and not necessarily over $\mathbb{Z}$. To show that no overflow has occurred, output amounts are also proved to lie in the correct range using a Bulletproofs+ range proof π [24]. What remains to be shown is that pseudo-output commitments are consistent with the input commitments. This is proved by showing, for each input j, the knowledge of $z_j = x_j - x'_j$ such that $C_j - C'_j = z_j \cdot G$.

Tags and Authorization. Both the knowledge of z_j and of the output key k^o_j are proved together with a CLSAG [39] linkable ring signature. Using ring signatures affords the sender k-anonymity, whereby outside observers cannot in general identify which UTXO was really spent with probability better than $1/k$.[10]

CLSAG (as used in Monero) has private keys of the form $(k^o, z) \in \mathbb{F}_q^2$ and public keys $(k^o \cdot G, z \cdot G) \in \mathbb{G}^2$.[11] The jth input to a transaction gets signed with (k^o_j, z_j), where k^o_j is the output key of the input note, and z_j is such that $z_j \cdot G = C_j - C'_j$. Note that $C_j - C'_j$ is supposed to be a commitment to the value zero (the H-components cancel). z_j is then its opening. To hide the true input, $k-1$ *decoy* inputs are chosen at random from the ledger. These are, in general, other users' notes (possibly even already spent). Their stealth addresses K^o_{ji} and output commitments C_{ji} are used in the CLSAG signature. Specifically, the decoy public keys become $\{(K^o_{ji}, C_{ji} - C'_j)\}_{i=1}^{k-1}$. The real public key $(K^o_j, C_j - C'_j)$ is then shuffled in at some random position $\ell \in \{1, \ldots, k\}$. The signature only shows that the transaction originator knows the secret key corresponding to *one* of the public keys involved. No efficient adversary can tell with non-negligible advantage over $1/k$ which public key was actually used to produce the signature [39]. Two *key images* are involved in the signature and revealed alongside it: a primary key image $\tilde{K} = k^o_j \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_j)$ and an auxiliary key image $\tilde{Z} = z_j \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_j)$ [46]. The primary key image $\tilde{K}$ plays the role of the note's tag: if double-spending is attempted, $\tilde{K}$ will be repeated, and thus reuse of k^o will be detected.

A more detailed description of the CLSAG signature scheme can be found in Appendix C. For now, let us summarize what a Monero transaction looks like. For simplicity, consider a single-input single-output transaction T that spends a UTXO $u \in \{u_1, \ldots, u_k\}$ and mints u'. Then

$$T = \left(\{u_1, \ldots, u_k\}, C', \sigma, \tilde{K}, \tilde{Z}, \pi, u'\right),$$

[10] With $k = 16$ as of this writing. See `https://xmrchain.net` (accessed: 30 August 2026). Notably, this simplified view ignores correlations between transactions that increase the distinguisher's advantage when the *cover set* is not chosen carefully [32, 72, 73]. This is not relevant for the present work.

[11] Goodell *et al.* [39] define CLSAG more generally for public keys $(Z_1, \ldots, Z_d) \in \mathbb{G}^d$ for some d. What is used in Monero corresponds to 2-CLSAG using their terminology.

with C' a pseudo-output commitment, σ a CLSAG ring signature [39], $\tilde{K}, \tilde{Z} \in \mathbb{G}$ the two key images, and π a Bulletproofs+ [24] range proof.

B.2 Zcash

Zcash is based on Zerocash [11], itself based on Zerocoin [52]. The present work focuses solely on Zcash's most recent *shielded protocol*,[12] which, as of this writing, is *Orchard* [40].[13] An Orchard note has a particularly simple form:

$$u = (\mathsf{cm}, \mathsf{epk}, C^{\mathsf{enc}}) \,,$$

where cm is a *note commitment* and the pair $(\mathsf{epk}, C^{\mathsf{enc}})$ is supposed to be a ciphertext in an ElGamal-based hybrid encryption scheme carrying the witnesses for the recipient.[14] The note commitment cm is to be computed as

$$\mathsf{cm} \leftarrow \mathsf{Hash}_{\mathbb{G}}(\mathsf{g_d}, \mathsf{pk_d}, v, \rho, \psi) + \mathsf{rcm} \cdot H \,,$$

where $\mathsf{g_d}$ and $\mathsf{pk_d}$ are components of the recipient's address, v is the note's amount, rcm and ψ are randomizers sampled by the sender, and ρ is the tag of one of the inputs to the transaction that minted the note (see below). When a note u is minted, its commitment cm is added as a leaf to a global append-only Merkle tree, called the *note commitment tree* [40, Section 3.8]. To spend u, one must prove that cm is present in the tree, that one knows the secret spending key associated to $\mathsf{g_d}$ and $\mathsf{pk_d}$, and that the amounts in the transaction balance overall. All this is proved in zero knowledge meaning in particular that the anonymity set for each spend is equal to at least the set of all outstanding UTXOs. Notably, this set is itself implicit as one cannot in general decide if a given note has already been spent or not.

Key Hierarchy. Let us now elaborate the simple idea from above, starting with the key hierarchy defined by Orchard. At the top of this hierarchy, we find the *spend-authorizing key* $\mathsf{ask} \in \mathbb{F}_q$ and the *full viewing key*

$$\mathsf{fvk} = (\mathsf{ak}, \mathsf{nk}, \mathsf{rivk}) \in \mathbb{G} \times \mathbb{F}_p \times \mathbb{F}_q \,.$$

In practice, ask, nk, and rivk are typically derived from a single master seed. For simplicity, we will assume they are sampled independently. The *spend-validating key* ak is derived as $\mathsf{ak} \leftarrow \mathsf{ask} \cdot G$. From the full viewing key one then derives three subsidiary viewing keys, namely, the *diversifier key* dk, the *incoming viewing key*

[12] Zcash also supports a transparent protocol alongside it, which is equivalent to Bitcoin and hence of no interest.

[13] On 28 July 2026, an upgrade called *Ironwood* launched that introduced a new "shielded pool" but it nevertheless continues to use the Orchard protocol.

[14] Analogous to (R, c) for Monero.

ivk, and the *outgoing viewing key* ovk as follows.[15] The incoming viewing key is derived as

$$\mathsf{ivk} \leftarrow \mathbf{x}\big(\mathsf{Hash}_{\mathbb{G}}(\mathsf{ak}, \mathsf{nk}) + \mathsf{rivk} \cdot H\big) \bmod q\,. \tag{2}$$

In Zcash, we assume $q > p$. Note that ivk is computed as an x-coordinate of an element of $\mathbb{G}$, itself an element of the base field $\mathbb{F}_p$ (here identified with $\mathbb{Z}/(p)$), and only then lifted to the (larger) scalar field $\mathbb{F}_q$. This is of some minor significance in the proof of Theorem 1. The outgoing viewing key ovk and the diversifier key dk are derived jointly as

$$\mathsf{dk} \| \mathsf{ovk} \leftarrow \mathsf{Hash}_{\{0,1\}^{4\lambda}}(\mathsf{rivk}, \mathsf{ak}, \mathsf{nk})\,, \tag{3}$$

where dk and ovk are each 2λ bits long, two halves of the output of the hash function.

Addressing Notes. From $(\mathsf{dk}, \mathsf{ivk})$ one finally derives a payment address. In fact, many so-called *diversified addresses* can be derived from a single $(\mathsf{dk}, \mathsf{ivk})$ pair. To create a new address, a user picks an index $i \in \{0,1\}^{88}$ and applies a pseudorandom permutation Enc keyed with dk to it to get a *diversifier*

$$\mathsf{d} \leftarrow \mathsf{Enc}_{\mathsf{dk}}(i)\,.$$

(We write Enc for the permutation, as it is implemented with the AES block cipher.) From the diversifier, a fresh generator of the group $\mathbb{G}$ is derived as

$$\mathsf{g_d} \leftarrow \mathsf{Hash}_{\mathbb{G}}(\mathsf{d})\,.$$

The diversified address corresponding to the index i is then the pair $(\mathsf{d}, \mathsf{pk_d})$, where

$$\mathsf{pk_d} = \mathsf{ivk} \cdot \mathsf{g_d}\,.$$

Recall that a note u gets addressed to $(\mathsf{d}, \mathsf{pk_d})$ by committing to $\mathsf{g_d}$ and $\mathsf{pk_d}$ in the note commitment cm. Additionally, one normally encrypts the witnesses needed later to spend u, such as the randomizers rcm and ψ under the Diffie-Hellman key $\mathsf{esk} \cdot \mathsf{pk_d}$ for uniform $\mathsf{esk} \leftarrow\!\!\$\ \mathbb{F}_q$, setting $\mathsf{epk} \leftarrow \mathsf{esk} \cdot \mathsf{g_d}$. This is the ciphertext C^{enc} attached to u.

Clearly, when u is minted honestly (*i.e.*, according to the specification), knowledge of ivk suffices to decrypt it. Consistency of C^{enc} is, however, not subject to the consensus protocol. The sender may choose to encrypt random data under a random key instead and only send the witnesses (ψ, rcm, v, etc.) out of band to the recipient [40, Section 4.20.1]. The holder of ivk in this case will not even be able to recognize that u credits $\mathsf{pk_d}$. Indeed, incoming notes are normally identified solely based on whether they successfully decrypt under ivk

[15] The Zcash specification [40] actually refers to the pair $(\mathsf{dk}, \mathsf{ivk})$ as the incoming viewing key. Since we will typically not be interested in dk, we shall for brevity reserve the term for ivk alone. This is also in line with the naming used in older versions of the Zcash protocol (Sapling).

or not, which such "malformed" u would not. Interestingly, an older version of the Zcash protocol, called Sapling, required the sender prove the ephemeral key epk was derived honestly, namely, prove they know esk [40]. Consistency of the ciphertext C^{enc}, however, was not verified.

When minting u, the sender may also choose to enable the holder of *their* (not the recipient's) outgoing viewing key ovk to see the details of u. This is simply done by encrypting $(\mathsf{pk_d}, \mathsf{esk})$ under ovk. The corresponding ciphertext, denoted C^{out}, is included in the transaction. This is again done solely at the sender's discretion [40, Section 4.20.1].

Actions. For the purposes of proving balance as well as tag derivation, the concept of *actions* will be important. An *action* is simply a spend-mint pair $(u_i, \hat{u}_j)$, where the note u_i is getting spent and, in the same transaction, the note $\hat{u}_j$ is getting minted. When creating a transaction, its originator must decide on how to match spends and mints. This is typically done at random.

For example, suppose we want to, in a single transaction, spend the UTXOs u_1, u_2, and u_3, and mint two new notes, say, $\hat{u}_1$ and $\hat{u}_2$. This is the very common scenario of aggregating funds from several UTXOs, here three, in order to make a payment in $\hat{u}_1$ and then send back change in $\hat{u}_2$. Since Zcash requires that spends and mints be paired together, and we are short one spend, we add a dummy one, call it $\hat{u}_3$, with value $v = 0$ addressed to a random recipient.[16] We then choose a random matching, *e.g.*, $\{(u_1, \hat{u}_3), (u_2, \hat{u}_2), (u_3, \hat{u}_1)\}$.

Balance. When proving balance, the transaction originator attaches a *value commitment* $\mathsf{cv}^{\mathsf{net}}$ to each action $(u, \hat{u})$. Let v be the amount attached to the input note u and let $\hat{v}$ be the amount attached to the output note $\hat{u}$. The transaction creator should sample the value commitment as

$$\mathsf{cv}^{\mathsf{net}} \leftarrow \mathsf{rcv} \cdot G + (v - \hat{v}) \cdot H \,, \tag{4}$$

for a random trapdoor $\mathsf{rcv} \leftarrow\!\!\$\, \mathbb{F}_q$.

Let $\mathsf{cv}_j^{\mathsf{net}} = \mathsf{rcv}_j \cdot G + (v_j - \hat{v}_j) \cdot H$ be the value commitment of the jth action and consider summing the commitments together:

$$\sum_j \mathsf{cv}_j^{\mathsf{net}} = \big(\sum_j \mathsf{rcv}_j\big) \cdot G + \big(\sum_j v_j - \sum_j \hat{v}_j\big) \cdot H \,.$$

If the transaction balances, *i.e.*, $\sum_j v_j = \sum_j \hat{v}_j$, then the H-component above will vanish, in which case the originator will know the discrete logarithm of $\sum_j \mathsf{cv}_j^{\mathsf{net}}$ to base G, namely, $\sum_j \mathsf{rcv}_j$. This knowledge is proved by signing the transaction to produce the so-called *binding signature* $\sigma_{\mathsf{binding}}$ [40, Section 4.14].[17]

[16] Dummy spends are also possible, see [40, Section 4.8.3] for details.

[17] We note here a complication one finds in the real world that is purposefully overlooked here. As per an earlier footnote, Zcash supports a transparent protocol alongside Orchard. When sending funds from the transparent pool to the Or-

Compare this with Monero, where the commitment trapdoors are sampled in a special way, so that the commitments sum to the identity element ∞ (*i.e.*, both the G- and H-components vanish). This direct algebraic relation is what causes problems when trying to prove Conjecture 1 and also limits the applicability of Theorem 4.

Tags and Authorization. When spending a note u, the transaction must reveal its tag (referred to as a *nullifier* in [40] or as a *serial number* in [11]). It is computed as follows:

$$\mathsf{nf} \leftarrow \mathsf{x}\big((\mathsf{Hash}_{\mathbb{F}_p}(\mathsf{nk}, \rho) + \psi \bmod p) \cdot K + \mathsf{cm}\big)\,, \tag{5}$$

with K one of the fixed generators of $\mathbb{G}$, with $\psi \in \mathbb{F}_p$ committed to in u's note commitment cm, $\rho \in \mathbb{F}_p$ fixed to be the tag of the note spent in the action that minted u, and nk the *nullifier deriving key* being part of the full viewing key fvk [40, Section 4.16].

To authorize spending u, one first randomizes the spend-authorizing key ask by sampling $\alpha \leftarrow\!\$\, \mathbb{F}_q$ and setting

$$\mathsf{rsk} \leftarrow \mathsf{ask} + \alpha\,.$$

One then signs the transaction with rsk and reveals the *randomized spend-validating key* $\mathsf{rk} = \mathsf{rsk} \cdot G$ [40, Section 4.15]. The link between rk and the address $\mathsf{pk_d}$ committed to in cm is proved as part of the zk-SNARK π attached to the transaction and elaborated below.

Finally, everything is tied together with a Halo2 [16] zk-SNARK π that proves, among some other minor things, that, for every action $(u, \hat{u})$,

- the randomized spend-validating key rk was correctly derived from ak, which in turn was used to derive ivk via Equation (2),
- the value commitment $\mathsf{cv}^{\mathsf{net}}$ was correctly computed as per Equation (4),
- the input amount v and the output amount $\hat{v}$ lie within the correct range and do not lead to an overflow in $\mathsf{cv}^{\mathsf{net}}$,
- the new note commitment is correct,
- the nullifier nf was computed correctly as per Equation (5),
- the old note commitment cm corresponds to an address $\mathsf{pk_d} = \mathsf{ivk} \cdot \mathsf{g_d}$, and
- cm was added in the past to the note commitment tree.

See [40, Section 4.18.4] for the full statement.

In summary, a (single-input single-output) Zcash transaction that spends a UTXO u and mints a new one u' looks like:

$$T = (\mathsf{rt}, \mathsf{cv}^{\mathsf{net}}, C^{\mathsf{out}}, \mathsf{nf}, \sigma, \mathsf{rk}, \sigma_{\mathsf{binding}}, \pi, u')\,,$$

chard/Ironwood pool (shielding) or *vice versa* (deshielding), one accounts for the public net change of supply v^{balance} in the Orchard/Ironwood pool with an extra commitment $v^{\mathsf{balance}} \cdot H$.

with rt the *anchor* corresponding to the root of the note commitment tree at some point in the past, $\mathsf{cv}^{\mathsf{net}}$ the value commitment, C^{out} the "outgoing" ciphertext, nf the tag of the input note u, σ a spend-authorizing signature under a randomized spend-validating key rk, π a Halo2 [16] zk-SNARK, and $\sigma_{\mathsf{binding}}$ a binding signature.

B.3 Firo

Firo is based on Lelantus Spark [43] which draws heavily on both Monero and Zcash for its design. Indeed, a Firo UTXO looks almost identical to a Monero UTXO. Specifically, it has the form:

$$u = (S, R, C, c)\,,$$

where $S, R, C \in \mathbb{G}$ and c is a ciphertext. As in Monero, C is a value commitment, *i.e.*, a Pedersen commitment to the amount attached to u, and R is an ephemeral key that is supposed to be used to decrypt c. S, on the other hand, is a commitment to u's serial number s. When spending u, the transaction will need to reveal a tag derived from s.

Key Hierarchy. Firo uses a similar key hierarchy to Zcash. At the top we find the spend-authorizing key

$$\mathsf{addr}_{\mathrm{sk}} = (s_1, s_2, r) \in \mathbb{F}_q^3\,,$$

sampled uniformly. From it, one derives the *full viewing key*

$$\mathsf{addr}_{\mathrm{full}} \leftarrow (s_1, s_2, D, P_2) \in \mathbb{F}_q^2 \times \mathbb{G}^2\,,$$

where $D = r \cdot G$ and $P_2 = s_2 \cdot F + D$. From $\mathsf{addr}_{\mathrm{full}}$, finally, one derives the *incoming viewing key*

$$\mathsf{addr}_{\mathrm{in}} \leftarrow (s_1, P_2) \in \mathbb{F}_q \times \mathbb{G}\,.$$

No outgoing viewing key is defined.

Addressing Notes. From the incoming viewing key one derives diversified addresses similarly as in Zcash. To create a new address, a user picks an index $i \in \mathbb{N}$ and encrypts it with s_1 to get a diversifier[18]

$$d \leftarrow \mathsf{Enc}_{s_1}(i)\,.$$

The diversifier then determines a fresh generator of $\mathbb{G}$ similarly as in Zcash. One sets

$$Q_1 \leftarrow s_1 \cdot \mathsf{Hash}_{\mathbb{G}}(d) \quad \text{and} \quad Q_2 \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(2, s_1, i) \cdot F + P_2\,.$$

[18] Jivanyan and Feickert [43] call i itself a diversifier, but this work shall adhere to the naming convention established by Zcash [40].

The diversified address corresponding to the index i is then the tuple (d, Q_1, Q_2).

To send a note u to (d, Q_1, Q_2), the sender samples a nonce $z \leftarrow\!\!\$\ \mathbb{F}_q$ and commits to a serial number by setting

$$S \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(3, z) \cdot F + Q_2 \,.$$

The recipient, given the nonce z, the diversifier d, and their full viewing key $\mathsf{addr}_{\text{full}}$ can compute this serial number as

$$s \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(3, z) + \mathsf{Hash}_{\mathbb{F}_q}(2, s_1, i) + s_2 \,,$$

where they find the index as $i \leftarrow \mathsf{Dec}_{s_1}(d)$ [43, Section 4.7]. Observe that

$$S = s \cdot F + D \,.$$

The value v is committed to using a trapdoor derived from the nonce z as well. Specifically, the sender sets

$$C \leftarrow v \cdot G + \mathsf{Hash}_{\mathbb{F}_q}(4, z) \cdot H \,.$$

The component Q_1 of the address is only used as part of the public-key encryption scheme used to send the witnesses on chain. Specifically, the nonce z, the diversifier d, and the amount v attached to u are normally supposed to be encrypted (using again an ElGamal-based hybrid encryption scheme) and attached to the note. The sender is supposed to compute an ephemeral key

$$R \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, z) \cdot \mathsf{Hash}_{\mathbb{G}}(d) \,,$$

and encrypt the note under $E \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, z) \cdot Q_1$:

$$c \leftarrow\!\!\$\ \mathsf{Enc}_E(v, d, z) \,.$$

Clearly, when u is minted honestly like this, then the knowledge of $\mathsf{addr}_{\text{in}}$ (indeed, s_1) suffices to decrypt it.

As in the case of Zcash and Monero, the consistency of the ciphertext c is not verified in any way by the consensus protocol. Also, successful decryption of c is the only way the holder of $\mathsf{addr}_{\text{in}}$ can identify that the note credits (d, Q_1, Q_2). If c is malformed and witnesses, say, z are sent out of band, then the holder of $\mathsf{addr}_{\text{in}}$ is completely oblivious as to whom u credits.

Sender Anonymity. To hide which UTXO u is being spent, Firo uses a technique similar to Monero, namely, the sender pulls in $k - 1$ decoys at random from the ledger and demonstrates that one out of the k UTXOs referenced is being spent. To this end, the sender includes in a transaction two "commitment offsets" [43, Section 4.8]: a *serial number commitment offset*

$$S' \leftarrow s \cdot F - \mathsf{Hash}_{\mathbb{F}_q}(5, s, D) \cdot H + D \tag{6}$$

and a *value commitment offset*

$$C' \leftarrow v \cdot G + \mathsf{Hash}_{\mathbb{F}_q}(6, s, D) \cdot H\,, \tag{7}$$

where s and v are, respectively, the serial number and the amount associated with u, and D is the component of the full viewing key $\mathsf{addr}_{\text{full}}$. Their purpose is analogous to Monero's pseudo-output commitment (see Appendix B.1). Recall that the serial number commitment S of u satisfies $S = s \cdot F + D$. Therefore, the sender is expected to know the discrete logarithm of $S - S'$ to base H, which is $\mathsf{Hash}_{\mathbb{F}_q}(5, s, D)$. Similarly, the sender should know the discrete logarithm of $C - C'$ to base H. They demonstrate the knowledge of both using a *parallel one-out-of-many proof* [43, Section 2.4] for the relation

$$\big\{\{S_i, V_i\}_{i=1}^{k} \subseteq \mathbb{G}^2, S', V' \in \mathbb{G}; \ell \in \mathbb{N}, s, v \in \mathbb{F}_q \mid 1 \leq \ell \leq k \text{ and } S_\ell - S' = s \cdot H \text{ and } V_\ell - V' = v \cdot H\big\}\,.$$

Note how the proof is functionally equivalent to the CLSAG ring signature from Monero (see Appendix C), in that it shows the prover knows the openings of the two commitments for some note in the cover set $\{u_1, \ldots, u_k\}$. Notably, however, because it is formulated as a non-interactive zero-knowledge proof, one can invoke its simulator to prove indistinguishability, which is not possible in the case of CLSAG (see Conjecture 1).

Balance. Balance is proved analogously to how it is done in Zcash, namely, the sender demonstrates the knowledge of the discrete logarithm of the aggregate value commitment $\sum_j C'_j - \sum_t \hat{C}_t$ to base H, where C'_j is the value commitment offset of the jth input and $\hat{C}_t$ is the value commitment of the tth output. In doing so, they demonstrate that the G-component has vanished, *i.e.*, the amounts balance modulo the group order. (Compare again with Monero, where both components vanish.) This is then coupled with a simple *range proof* [43, Section 2.7] to account for the possibility of overflow.

Tags and Authorization. To spend u, its owner computes a tag

$$N \leftarrow \frac{1}{s} \cdot (K - D)\,, \tag{8}$$

where s is u's serial number, D is the component of the full viewing key $\mathsf{addr}_{\text{full}}$, and K is a fixed generator of $\mathbb{G}$. They reveal N and prove that it is correctly computed, namely, that it is consistent with the serial number commitment offset S'. They also prove to know r such that $D = r \cdot G$. This is achieved with a *modified Chaum-Pedersen proof* [43, Section 2.3] for the relation

$$\big\{\{S'_j, N_j\}_{j=1}^{w} \subseteq \mathbb{G}^2; \{x_j, y_j, z_j\}_{j=1}^{w} \subseteq \mathbb{F}_q^3 \mid (\forall j \in \{1, \ldots, w\})(S'_j = x_j \cdot F + y_j \cdot G + z_j \cdot H \text{ and } K = x_j \cdot N_j + y_j \cdot G)\big\}\,,$$

which batches the above claims across all w inputs to a transaction.

In summary, a single-input single-output Firo transaction that spends a UTXO $u \in \{u_1, \ldots, u_k\}$ and mints u' has the format:

$$T = \left(\{u_1, \ldots, u_k\}, S', C', N, \pi_{\text{par}}, \pi_{\text{chaum}}, \pi_{\text{bal}}, \pi_{\text{rp}}, u'\right),$$

where S' and C' are the commitment offsets defined above, N is the input note u's tag, π_{par} is the parallel one-out-of-many proof showing that S' and C' are consistent with some u_ℓ, π_{chaum} is the modified Chaum-Pedersen proof showing consistency of N and spend authority (knowledge of r), π_{bal} is a Schnorr proof demonstrating balance, and π_{rp} is a range proof showing that the amount attached to u' is in the correct range.

C CLSAG in Monero

This section elaborates the CLSAG ring signature scheme as used in Monero.

Consider a Monero transaction T and focus on its jth input denoted u. Let $\{u_1^{(j)}, \ldots, u_k^{(j)}\}$ be the cover set for this input with the real input at position ℓ. This means that $u_\ell^{(j)} = u$ and the remaining $u_i^{(j)}$ for $i \neq \ell$ were sampled from the Monero ledger. Parse each note in the cover set as

$$u_i^{(j)} = (K_{ji}^o, R_{ji}, C_{ji}, c_{ji}).$$

The originator of T samples a pseudo-output commitment C_j' for this jth input as described in Appendix B.1, so that they know z_j such that $z_j \cdot G = C_{j\ell} - C_j'$. The CLSAG private key becomes $(k_j^o, z_j) \in \mathbb{F}_q^2$, where k_j^o is the output key such that $K_{j\ell}^o = k_j^o \cdot G$. Write $Z_{ji} = C_{ji} - C_j'$. The decoy ring becomes $\{(K_{ji}^o, Z_{ji})\}_{i=1}^k$.

The CLSAG signing and verification procedures are presented in Figures 2 and 3, respectively. The key generation algorithm is straightforward and omitted for brevity: it samples k^o and z uniformly from $\mathbb{F}_q$ and returns $\mathsf{sk} = (k^o, z)$ and $\mathsf{pk} = (k^o \cdot G, z \cdot G)$. Observe in Figure 2 that, out of the k nonces $r_1, \ldots, r_k \in \mathbb{F}_q$ comprising a CLSAG signature, only one is derived deterministically, namely, r_ℓ. The remaining ones are drawn uniformly at random from $\mathbb{F}_q$. This is the basis for the subliminal channel studied in [25] and alluded to in Section 4.1.

For completeness, let us recall the *linkable anonymity* notion that CLSAG signatures are supposedly shown in [39] to satisfy. It is implicitly assumed by the community that this implies spend indistinguishability for Monero (Conjecture 1). As explained in Section 3, however, this is not the case in general. Furthermore, we note that the definition contains an error that is overlooked in the proof due Goodell *et al.* [39]. Indeed, the signing oracle $\mathcal{O}_{\mathsf{Sign}}$ conflates two events into a single check and answers with an honest signature under the key pk requested whenever

$$\{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\} \not\subseteq Q \quad \text{or} \quad \mathsf{pk} \notin \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\}.$$

The "or" is the problem here. A trivial adversary wins the experiment with probability negligibly close to 1 as follows. First, query $\mathcal{O}_{\mathsf{Sign}}$ on any m, $Q =$

$\mathsf{Sign}(\mathsf{sk}, m, \{\mathsf{pk}_1, \ldots, \mathsf{pk}_k\})$

1 : Let ℓ be such that $\mathsf{sk} \cdot G = \mathsf{pk}_\ell$
2 : $(k^o_j, z_j) \leftarrow \mathsf{sk}$
3 : $Q \leftarrow \mathsf{pk}_1 \| \ldots \| \mathsf{pk}_k$
4 : **foreach** $i \in \{1, \ldots, k\}$ **do**
5 : $\quad (K^o_{ji}, Z_{ji}) \leftarrow \mathsf{pk}_i$
6 : **endforeach**
7 : $\tilde{K} \leftarrow k^o_j \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{j\ell})$
8 : $\tilde{Z} \leftarrow z_j \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{j\ell})$
9 : **foreach** $i \in \{1, \ldots, k\} \setminus \{\ell\}$ **do**
10 : $\quad r_i \leftarrow\!\!\$\ \mathbb{F}_q$
11 : $\quad W_i \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, Q, \tilde{K}, \tilde{Z}) \cdot K^o_{ji} + \mathsf{Hash}_{\mathbb{F}_q}(2, Q, \tilde{K}, \tilde{Z}) \cdot Z_{ji}$
12 : **endforeach**
13 : $w_\ell \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, Q, \tilde{K}, \tilde{Z}) \cdot k^o_j + \mathsf{Hash}_{\mathbb{F}_q}(2, Q, \tilde{K}, \tilde{Z}) \cdot z_j$
14 : $\tilde{W} \leftarrow w_\ell \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{j\ell})$
15 : $\alpha \leftarrow\!\!\$\ \mathbb{F}_q$
16 : $c_{(\ell \bmod k)+1} \leftarrow \mathsf{Hash}_{\mathbb{F}_q}\big(Q, m, \alpha \cdot G, \alpha \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{j\ell})\big)$
17 : **for** $i = (\ell \bmod k) + 1, (\ell + 1 \bmod k) + 1, \ldots, (\ell + k - 2 \bmod k) + 1$ **do**
18 : $\quad c_{(i \bmod k)+1} \leftarrow \mathsf{Hash}_{\mathbb{F}_q}\big(Q, m, (r_i \cdot G + c_i \cdot W_i), (r_i \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{ji}) + c_i \cdot \tilde{W})\big)$
19 : **endfor**
20 : $r_\ell \leftarrow \alpha - c_\ell \cdot w_\ell \bmod q$
21 : $\sigma = (c_1, r_1, \ldots, r_k)$
22 : **return** $(\sigma, \tilde{K}, \tilde{Z})$

Fig. 2. Signing procedure of the 2-CLSAG ring signature scheme as used in Monero

$\mathsf{Verify}(\{\mathsf{pk}_1, \ldots, \mathsf{pk}_k\}, m, \sigma, \tilde{K}, \tilde{Z})$

$Q \leftarrow \mathsf{pk}_1 \| \ldots \| \mathsf{pk}_k$
$(c_1, r_1, \ldots, r_k) \leftarrow \sigma$
foreach $i \in \{1, \ldots, k\}$ **do**
  $(K^o_{ji}, Z_{ji}) \leftarrow \mathsf{pk}_i$
  $W_i \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, Q, \tilde{K}, \tilde{Z}) \cdot K^o_{ji} + \mathsf{Hash}_{\mathbb{F}_q}(2, Q, \tilde{K}, \tilde{Z}) \cdot Z_{ji}$
endforeach
$\tilde{W} \leftarrow \mathsf{Hash}_{\mathbb{F}_q}(1, Q, \tilde{K}, \tilde{Z}) \cdot \tilde{K} + \mathsf{Hash}_{\mathbb{F}_q}(2, Q, \tilde{K}, \tilde{Z}) \cdot \tilde{Z}$
for $i = 1, \ldots, k-1$ **do**
  $c_{i+1} \leftarrow \mathsf{Hash}_{\mathbb{F}_q}\big(Q, m, (r_i \cdot G + c_i \cdot W_i), (r_i \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{ji}) + c_i \cdot \tilde{W})\big)$
endfor
$c'_1 \leftarrow \mathsf{Hash}_{\mathbb{F}_q}\big(Q, m, (r_k \cdot G + c_k \cdot W_k), (r_k \cdot \mathsf{Hash}_{\mathbb{G}}(K^o_{jk}) + c_k \cdot \tilde{W})\big)$
return $c_1 = c'_1$

Fig. 3. Verification procedure of the 2-CLSAG ring signature scheme as used in Monero

$\{\mathsf{pk}_{i_0}\}$, and $\mathsf{pk} = \mathsf{pk}_{i_0}$. Since $\mathsf{pk}_{i_1} \notin Q$ but $\mathsf{pk} = \mathsf{pk}_{i_0}$, this triggers the if-branch in $\mathcal{O}_{\mathsf{Sign}}$. The oracle then produces a signature under pk as well as, more crucially, the linking tag $\tilde{K}$. The adversary may now query the oracle again with $Q = \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\}$ and compare the tag obtained in this query with $\tilde{K}$. If the two are the same, output 0; otherwise, output 1. The original definition due to Goodell *et al.* [39] is therefore unsatisfiable by *any* linkable ring signature scheme. We fix their security experiment in Figure 5, where we have oracle sign honestly only when $\mathsf{pk} \notin \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\}$ and abort in the "leaky" case of $\mathsf{pk} \in \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\} \not\subseteq Q$.

Definition 3 (Chosen-Target Linkable Anonymity). *A linkable ring signature scheme* $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* linkably anonymous *if, for all efficient algorithms* $\mathcal{A}$ *and all* k*, there exists a negligible function* ε *such that*

$$\Pr[\mathsf{LinkAnon}^k_{\mathcal{A},\Sigma}(\lambda) = 1] \leq \frac{1}{2} + \varepsilon(\lambda)\,,$$

where the experiment $\mathsf{LinkAnon}^k_{\mathcal{A},\Sigma}$ *is presented in Figure 4 (and fixed in Figure 5), and the probability is taken over the random coin tosses of all algorithms involved.*

Much of the proof of CLSAG's linkable anonymity [39, Theorem 4] can be salvaged. However, the original proof falsely claimed that, in the DDH simulation, the adversary $\mathcal{A}$ "can do no better than random chance at determining [the secret bit]." This is only true if the oracle never gives honest answers to queries about either of the challenge keys pk_{i_0} and pk_{i_1}, *i.e.*, the theorem only holds under the fixed definition with experiment in Figure 5.

$\mathsf{LinkAnon}^{k}_{\mathcal{A},\Sigma}(\lambda)$

1 : $b \leftarrow\!\$\ \{0,1\}$
2 : **foreach** $i \in \{1,\ldots,k\}$ **do**
3 : $\quad(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
4 : **endforeach**
5 : $S \leftarrow \{\mathsf{pk}_i\}_{i=1}^{k}$
6 : $i_0, i_1, \mathsf{st} \leftarrow\!\$\ \mathcal{A}(\{\mathsf{pk}_i\}_{i=1}^{k})$
7 : $b' \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{st})$
8 : **return** $b = b'$

$\mathcal{O}_{\mathsf{Sign}}(m, Q, \mathsf{pk})$

1 : **if** $\mathsf{pk} \notin Q \cap S$ **then**
2 : $\quad$**return** $\perp$
3 : **endif**
4 : Let i be such that $\mathsf{pk}_i = \mathsf{pk}$
5 : **if** $\{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\} \not\subseteq Q$
$\quad\quad \vee\ \mathsf{pk} \notin \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\}$ **then**
6 : $\quad(\sigma, \tilde{K}, \tilde{Z}) \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}_i, m, Q)$
7 : **else**
8 : $\quad$Let c be such that $\mathsf{pk}_{i_c} = \mathsf{pk}$
9 : $\quad c' \leftarrow c \oplus b$
10 : $\quad(\sigma, \tilde{K}, \tilde{Z}) \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}_{i_{c'}}, m, Q)$
11 : **endif**
12 : **return** $(\sigma, \tilde{K}, \tilde{Z})$

Fig. 4. Chosen-target linkable anonymity experiment quoted from [39]

$\mathsf{LinkAnon}^{k}_{\mathcal{A},\Sigma}(\lambda)$

1 : $b \leftarrow\!\$\ \{0,1\}$
2 : **foreach** $i \in \{1,\ldots,k\}$ **do**
3 : $\quad(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
4 : **endforeach**
5 : $S \leftarrow \{\mathsf{pk}_i\}_{i=1}^{k}$
6 : $i_0, i_1, \mathsf{st} \leftarrow\!\$\ \mathcal{A}(\{\mathsf{pk}_i\}_{i=1}^{k})$
7 : $b' \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{st})$
8 : **return** $b = b'$

$\mathcal{O}_{\mathsf{Sign}}(m, Q, \mathsf{pk})$

1 : **if** $\mathsf{pk} \notin Q \cap S$ **then**
2 : $\quad$**return** $\perp$
3 : **endif**
4 : Let i be such that $\mathsf{pk}_i = \mathsf{pk}$
5 : **if** $\mathsf{pk} \notin \{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\}$ **then**
6 : $\quad(\sigma, \tilde{K}, \tilde{Z}) \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}_i, m, Q)$
7 : $\quad$**return** $(\sigma, \tilde{K}, \tilde{Z})$
8 : **elseif** $\{\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1}\} \not\subseteq Q$ **then**
9 : $\quad$**return** $\perp$
10 : **else**
11 : $\quad$Let c be such that $\mathsf{pk}_{i_c} = \mathsf{pk}$
12 : $\quad c' \leftarrow c \oplus b$
13 : $\quad(\sigma, \tilde{K}, \tilde{Z}) \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}_{i_{c'}}, m, Q)$
14 : $\quad$**return** $(\sigma, \tilde{K}, \tilde{Z})$
15 : **endif**

Fig. 5. Chosen-target linkable anonymity experiment fixed

D Proofs

This section revisits the claims about Zcash and Firo having indistinguishable spends relative to incoming viewing keys and formally establishes Theorems 1 and 2. Both proofs proceed in a similar fashion, using a standard technique of defining a sequence of hybrids starting with $\mathsf{G}_0 = \mathsf{SpendInd}^{0,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}$ and ending with $\mathsf{SpendInd}^{1,n,V}_{\Pi,\mathsf{Coinbase},\mathcal{S},\mathcal{D}}$. Each consecutive pair of games will be indistinguishable to efficient adversaries. Write p_i for the probability that $\mathcal{D}$ returns 1 in the ith hybrid G_i and write, for brevity, $\mathsf{G}_i \mapsto \mathsf{G}_j$ to denote the changes introduced to game G_i to arrive at game G_j. We shall model all hash functions as random oracles and abuse notation by writing, say, ivk to interchangeably mean the label mapping an owned UTXO to the incoming viewing key and the incoming viewing key itself, and similarly for other viewing keys. Finally, for clarity of exposition, we shall consider single-input single-output transactions only. The results are easily extended to transactions with multiple inputs or outputs if one wishes to restrict the spender algorithm $\mathcal{S}$ further (to account for, *e.g.*, heuristic analyses [30]).

D.1 Proof of Theorem 1

Proof (of Theorem 1). Recall from Appendix B.2 that a single-input single-output Zcash transaction is of the form

$$T = (\mathsf{rt}, \mathsf{cv}^{\mathsf{net}}, C^{\mathsf{out}}, \mathsf{nf}, \sigma, \mathsf{rk}, \sigma_{\mathsf{binding}}, \pi, u')\,,$$

with rt, $\mathsf{cv}^{\mathsf{net}}$, C^{out}, and $\sigma_{\mathsf{binding}}$ always independent of the input note, and also σ and rk independent of the input note unless the link between rk and ak is shown in the Halo2 [16] zk-SNARK π. The proof follows by leveraging the zero-knowledge property of Halo2 to simulate π and then decouple also the tag nf (and with it u' which uses nf as a randomizer in its note commitment) from the input note. Construct the spender algorithm $\mathcal{S}$ to always set C^{out} to an encryption of a random plaintext under a random key independent of the input note [40, Section 4.20.1]. Other than that, let $\mathcal{S}$ construct both T_0 and T_1 honestly, according to the Zcash protocol [11,40], with T_0 only spending u_{i_0} and T_1 only spending u_{i_1}. Therefore $g = g_0 \wedge g_1 = 1$ independently of b and the random coins tossed by $\mathcal{S}$.

Fix $b = 0$ and consider a modification to the original game, $\mathsf{G}_0 \mapsto \mathsf{G}_1$, whereby, instead of running Setup to generate the parameters pp honestly, we generate the Halo2 zk-SNARK [16] parameters with trapdoors (namely, sample the generators G, H, and K with known discrete logarithm relations), such that any argument, even for a false statement, can be simulated by $\mathcal{S}$. Since Halo2 has perfect zero knowledge [16], we get $p_0 = p_1$.

Games 1–3*: Making the Keys Independent.* Let $\mathsf{G}_1 \mapsto \mathsf{G}_2$ be that any incoming viewing key is derived *not* from the full viewing key $\mathsf{fvk} = (\mathsf{ak}, \mathsf{nk}, \mathsf{rivk})$ as in

Equation (2) but instead from an independent and uniformly random point $P_1 \leftarrow\!\!\$\, \mathbb{G}$ as

$$\mathsf{ivk} \leftarrow \mathbf{x}(P_1) \bmod q\,, \tag{9}$$

with the proof of address integrity [40, Section 4.18.4] simulated. It is straightforward to show that, in the random oracle model, the advantage in distinguishing between ivk computed as in Equation (2) and as in Equation (9) is bounded by the probability of the distinguisher querying the random oracle $\mathsf{Hash}_{\mathbb{G}}$ on $(\mathsf{ak}, \mathsf{nk}) \in \mathbb{G} \times \mathbb{F}_p$ (see [40, Section 4.2.3]), for either u_{i_0} or u_{i_1}, which is bounded by $\mathsf{poly}(\lambda)/2^\lambda$ for a polynomial-time distinguisher. The same bound then applies to $|p_1 - p_2|$. Next, let $\mathsf{G}_2 \mapsto \mathsf{G}_3$ be that the dk and ovk are also sampled independently at random instead of being computed as in Equation (3). By a similar argument as in the previous step, we have $|p_2 - p_3| \leq \mathsf{poly}(\lambda)/2^\lambda$. Note that, at this point, $\mathsf{ivk}(\bar{u}_{i_0})$ and $\mathsf{ovk}(\bar{u}_{i_0})$ are independent of any other key associated with the UTXO u_{i_0} being spent in T_0. The same can be said about u_{i_1} being spent in T_1.

Games 4–7*: Making the Tags Independent.* Write $\mathsf{nf} \in \mathbb{F}_p$ for the tag (nullifier) of u_{i_0} as revealed in T_0 returned by $\mathcal{S}$. It is computed (according to protocol) as in Equation (5) with $\rho, \psi \in \mathbb{F}_p$ and $\mathsf{cm} \in \mathbb{G}$ all associated with u_{i_0}, $K \in \mathbb{G}$ fixed by the consensus protocol, and nk the *nullifier deriving key* attached to $\bar{u}_{i_0}$ [40, Section 4.16]. In particular, ρ, ψ and cm are known to the dictator in possession of the incoming viewing keys for both challenge UTXOs, *i.e.*, $\mathsf{ivk}(\bar{u}_{i_0})$ and $\mathsf{ivk}(\bar{u}_{i_1})$. The nullifier is, at this point, the last thing "connecting" T_0 to the UTXO it spends. Let $\mathsf{G}_3 \mapsto \mathsf{G}_4$ be that, instead of computing the nullifier honestly, $\mathcal{S}$ sets it to $\mathsf{nf} \leftarrow \mathbf{x}(\alpha \cdot K + \mathsf{cm})$ for $\alpha \leftarrow\!\!\$\, \{0, \ldots, p-1\}$ uniformly random. $|p_3 - p_4|$ is again bounded by the probability of querying a random oracle, this time, $\mathsf{Hash}_{\mathbb{F}_p}$ on (nk, ρ), which itself is bounded by $\mathsf{poly}(\lambda)/2^\lambda$. Let $\mathsf{G}_4 \mapsto \mathsf{G}_5$ be the same transformation applied to u_{i_1} and T_1. We likewise have $|p_4 - p_5| \leq \mathsf{poly}(\lambda)/2^\lambda$.

Now, for $\mathsf{G}_5 \mapsto \mathsf{G}_6$, compute the nullifier of u_{i_0} as $\mathsf{nf} \leftarrow \mathbf{x}(P_2)$ instead, for $P_2 \leftarrow\!\!\$\, \mathbb{G}$ uniformly random. Clearly, nf is now independent of u_{i_0} and $|p_5 - p_6|$ is bounded by the statistical distance Δ between a uniform distribution on $\mathbb{G}$ and a uniform distribution on $X = \{\alpha \cdot K + \mathsf{cm} \mid \alpha \in \{0, \ldots, p-1\}\} \subset \mathbb{G}$, namely,

$$\Delta = \frac{1}{2} \sum_{P \in \mathbb{G}} \left| \Pr[\alpha \cdot K + \mathsf{cm} = P \mid \alpha \leftarrow\!\!\$\, \{0, \ldots, p-1\}] - \Pr[P_2 = P \mid P_2 \leftarrow\!\!\$\, \mathbb{G}] \right|.$$

Recall from Appendix B.2 that, for Zcash, we have $q > p$ to obtain:

$$\Delta = \frac{1}{2} \sum_{P \in X} \left| \frac{1}{p} - \frac{1}{q} \right| + \frac{1}{2} \sum_{P \in \mathbb{G} \setminus X} \frac{1}{q} = \frac{p}{2} \cdot \frac{q-p}{pq} + \frac{q-p}{2} \cdot \frac{1}{q} = \frac{q-p}{q}\,,$$

which is negligible by the Hasse bound. Specifically, we have $|q| = 2\lambda$, *i.e.*, $2^{2\lambda-1} \leq q \leq 2^{2\lambda}$, and

$$q - p \leq 2^{\lambda+2}\,,$$

from which we conclude that

$$|p_5 - p_6| \leq \Delta = \frac{q-p}{q} \leq \frac{2^{\lambda+2}}{2^{2\lambda-1}} = 2^{-\lambda+3}\,.$$

Let $\mathsf{G}_6 \mapsto \mathsf{G}_7$ be the same transformation applied to u_{i_1} and T_1. We have again that $|p_6 - p_7| \leq 2^{-\lambda+3}$.

Games 8–15*: Unwinding the Changes.* At this point, everything in T_0 and T_1 is independent of u_{i_0} and u_{i_1} (including the spend-authorizing keys used to sign T_0 and T_1). The transactions are therefore interchangeable. To say that T_0 now "spends" u_{i_1} and T_1 "spends" u_{i_0} in G_8 is merely a conceptual change with $p_7 = p_8$. This change is also equivalent to setting $b = 1$.

Running the above transformations in reverse, with $b = 1$, and defining games $\mathsf{G}_9, \ldots, \mathsf{G}_{15}$ accordingly, we get that

$$|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]| = |p_0 - p_{15}| \leq \frac{\mathsf{poly}(\lambda)}{2^\lambda}\,,$$

which is negligible. □

D.2 Proof of Theorem 2

Proof (of Theorem 2). Recall from Appendix B.3 that a single-input single-output Firo transaction is of the form

$$T = (\{u_1, \ldots, u_k\}, S', C', N, \pi, u')\,,$$

where the proofs

$$\pi = (\pi_{\mathrm{par}}, \pi_{\mathrm{chaum}}, \pi_{\mathrm{bal}}, \pi_{\mathrm{rp}})$$

will be simulated in the intermediate hybrids, which will also make the output note u' independent of the input. Also, any spender algorithm $\mathcal{S}$ that hopes to satisfy Definition 1 must clearly include both u_{i_0} and u_{i_1} in the cover set $\{u_1, \ldots, u_k\}$. It will therefore suffice to reason about indistinguishability of S', C', and N for a transaction T_0 spending u_{i_0} and T_1 spending u_{i_1}. This closely follows the proof of ledger indistinguishability (LIND) in [43], except there it suffices to say that the tags N are computed via a pseudorandom function of [31]. This simple argument no longer holds when the dictator gets extra information in the form of the incoming viewing keys $\mathsf{addr}_{\mathrm{in}}$.

Consider a spender algorithm $\mathcal{S}$ that produces T_0 and T_1 honestly except that, when computing the offsets S' and C', it samples $\alpha, \beta \leftarrow\!\$\ \mathbb{F}_q$ and sets

$$S' \leftarrow s \cdot F - \alpha \cdot H + D \quad \text{and} \quad C' \leftarrow v \cdot G + \beta \cdot H\,.$$

The specification [43, Section 4.8] tries de-randomizing this process and instead prescribes deriving α and β via a hash function as $\mathsf{Hash}_{\mathbb{F}_q}(5, s, D)$ and $\mathsf{Hash}_{\mathbb{F}_q}(6, s, D)$, respectively (see Equations (6) and (7)). This is, however, never

verified by the consensus protocol, so we might as well sample the scalars randomly. The transactions T_0 and T_1 returned by $\mathcal{S}$ spend, respectively, u_{i_0} and u_{i_1} only, ensuring that $g = g_0 \wedge g_1 = 1$ independently of b and the random coins tossed by $\mathcal{S}$.

Fix $b = 0$ and consider a modification to the original game, $\mathsf{G}_0 \mapsto \mathsf{G}_1$, whereby, instead of running Setup to generate the parameters pp honestly, we generate the parameters for the proofs π with trapdoors, such that any argument, even for a false statement, can be simulated by $\mathcal{S}$.[19] We have $p_0 = p_1$, as the proofs can be simulated perfectly. Let $\mathsf{G}_1 \mapsto \mathsf{G}_2$ be that, for each of T_0 and T_1, S' is sampled by $\mathcal{S}$ uniformly at random from $\mathbb{G}$ instead of being computed as above. This is merely a conceptual change, as S' is uniformly random and independent of D in either case. We therefore have $p_1 = p_2$.

Let $\mathsf{G}_2 \mapsto \mathsf{G}_3$ be that the tag N of u_{i_0} (as revealed in T_0) is chosen uniformly at random from $\mathbb{G}$ instead of being computed as in Equation (8). By the Decisional Diffie-Hellman Inverse (DDHI) assumption [19,31], there exists a negligible function $\varepsilon_{\mathrm{ddhi}}$ such that

$$|p_2 - p_3| \leq n \cdot \varepsilon_{\mathrm{ddhi}} \,. \tag{10}$$

Indeed, otherwise, given a DDHI triple $(B, A = a \cdot B, \tilde{A}) \in \mathbb{G}^3$, one could simulate $\mathcal{D}$'s view as follows and use it to decide if $\tilde{A} = (1/a) \cdot B$ or if it was sampled independently from $\mathbb{G}$. Sample $s_1, r \leftarrow\!\!\$\ \mathbb{F}_q$, parts of the spending key $\mathsf{addr}_{\mathrm{sk}}$ associated to u_{i_0}, honestly and set $D \leftarrow r \cdot G$. Sample $\gamma \leftarrow\!\!\$\ \mathbb{F}_q$ and produce the generators K and F as

$$K \leftarrow B + D \quad \text{and} \quad F \leftarrow \gamma \cdot B \,.$$

Set the tag of T returned by $\mathcal{S}$ to $N \leftarrow \tilde{A}$ and the serial number commitment in u_{i_0} to

$$S \leftarrow \gamma \cdot A + D \,.$$

We thus have $a \cdot F = S - D$, so a corresponds to the serial number s of u_{i_0} in the reduction. Sample uniform $w \leftarrow\!\!\$\ \mathbb{F}_q$ and program the random oracles to have

$$\mathsf{Hash}_{\mathbb{F}_q}(3, z) + \mathsf{Hash}_{\mathbb{F}_q}(2, s_1, i) = w \,,$$

for z the random nonce in u_{i_0} and set $P_2 \leftarrow S - w \cdot F$. To then simulate any other UTXO $\hat{u}$ under the control of $\mathsf{addr}_{\mathrm{sk}}$, sample the nonce $\hat{z}$ honestly, sample $\hat{w} \leftarrow\!\!\$\ \mathbb{F}_q$, and program the random oracles to have

$$\mathsf{Hash}_{\mathbb{F}_q}(3, \hat{z}) + \mathsf{Hash}_{\mathbb{F}_q}(2, s_1, i) = \hat{w} \,.$$

Finally, set the serial number commitment to

$$\hat{S} \leftarrow \hat{w} \cdot F + P_2 \,.$$

[19] Since the balance proof π_{bal} is just a Schnorr-like proof of knowledge of multiple discrete logarithms to the same base, to simulate it, we instead program the random oracle responses.

If $\tilde{A} = (1/a) \cdot B$, then the view of $\mathcal{D}$ is consistent with G_2; otherwise, it is consistent with G_3. Indeed, in the first case, $N = \tilde{A} = (1/a) \cdot B = (1/a) \cdot (K - D)$, and a corresponds to the serial number of u_{i_0}, since $a \cdot F = S - D$. We conclude that a $\mathcal{D}$ telling apart the two games can be leveraged to break the DDHI assumption, and so the games are indistinguishable. The factor of n in Equation (10) comes from the fact that our reduction must guess ahead of time the index i_0 that $\mathcal{D}$ is going to return.

Let $\mathsf{G}_3 \mapsto \mathsf{G}_4$ be the same transformation applied to u_{i_1}, namely, let us sample the tag N revealed in T_1 uniformly at random in G_4. By the same argument, we have $|p_3 - p_4| \leq n \cdot \varepsilon_{\text{ddhi}}$.

At this point, everything in T_0 and T_1 is independent of u_{i_0} and u_{i_1}, and the associated viewing keys $\mathsf{addr}_{\text{in}}(\bar{u}_{i_0})$ and $\mathsf{addr}_{\text{in}}(\bar{u}_{i_1})$. To say that T_0 now "spends" u_{i_1} and T_1 "spends" u_{i_0} in G_5 is another conceptual change with $p_4 = p_5$. It is also equivalent to setting $b = 1$. Running the above transformations in reverse, with $b = 1$, and defining games $\mathsf{G}_6, \ldots, \mathsf{G}_9$ accordingly, we get that

$$|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]| = |p_0 - p_9| \leq 4 \cdot n \cdot \varepsilon_{\text{ddhi}}\,,$$

which is negligible. □