Anamorphic Monero Transactions: the Threat of Bypassing Anti-Money Laundering Laws

Adrian Cinal[0009−0001−9789−0470], Przemysław Kubiak[0000−0002−1349−093X], Mirosław Kutyłowski[0000−0003−3192−2430], and Gabriel Wechta[0009−0009−8560−5300]

NASK National Research Institute, Warsaw, Poland
{adrian.cinal,przemyslaw.kubiak,
miroslaw.kutylowski,gabriel.wechta}@nask.pl

Abstract. In this paper, we analyze the clash between privacy-oriented cryptocurrencies and emerging legal frameworks for combating financial crime, focusing in particular on the recent European Union regulations. We analyze Monero, a leading "privacy coin" and a major point of concern for law enforcement, and study the scope of due diligence that must be exercised under the new law with regard to Monero trading platforms and how it translates to the technical capabilities of the Monero protocol. We both recognize flaws in the legislation and identify technical pitfalls threatening either the effective compliance of, say, Monero exchanges or the anonymity endeavour of Monero itself. Of independent interest is that we turn to anamorphic cryptography (marking one of the first practical applications of the concept) and leverage it to build a hidden transaction layer embedded in the Monero blockchain that obfuscates illegal money flow and circumvents transaction-level attempts at enforcing the EU law.

Keywords: Privacy coins · Anamorphic cryptography · Money laundering.

1 Introduction

In the last decade, we have witnessed an emergence of cryptocurrency markets, and, although some level of maturity of the technology has been achieved, the upcoming years may still bring substantial changes to the crypto ecosystem. These will be driven by the growing concern about applications of cryptocurrencies to money laundering, illegal financial transfers, evading sanctions, etc. In particular, reports such as [8] suggest that anonymous cryptocurrencies, so-called "privacy coins," are increasingly being used by criminals. A possible response of state authorities to this threat might be a general ban on cryptocurrency trade; see, e.g., the decision of the People's Bank of China in 2021 outlawing virtual currency-related business activities. Before rushing to prohibition, however, opportunities for the economy, related to the use of cryptocurrencies, should be taken into account. For this reason, the European Union aims to find a compromise between strict rules eliminating pathology in cryptocurrency trade and keeping the market open for innovations. Unfortunately, building a simple legal framework for the extremely complex cryptocurrency ecosystem (and the advanced cryptography involved in it) is a challenging task. Regardless of good intentions, regulating the market may fall short of the set goals, or worse: may have adverse effects.

1.1 Paper Contributions and Outline

We study the clash between privacy-preserving technologies as employed in cryptocurrencies and the law combating illegal financial trade. We specifically focus on the efforts of the regulators in Europe reflected in the recent crypto-assets regulation, MiCA [3], and the anti-money laundering directive (AML) [2], but our core conclusions transfer to similar legislation elsewhere. We study the case of Monero (XMR), hailed as "the leading cryptocurrency focused on private and censorship-resistant transactions,"[1] and attempt to answer the following research questions:

RQ1: Does the law discount privacy coins like Monero altogether, acknowledge and outlaw them, or open avenues for compliance without completely sacrificing the anonymity promises of the technology?

RQ2: What are the technical solutions available in Monero today to comply with the AML directive and MiCA and not sacrifice all anonymity?

RQ3: Letter of the law notwithstanding, can provably air-tight controls be implemented for Monero today that could block illegal transactions and, if so, at what cost to user privacy?

As for **RQ1**, we argue in Section 2.1 that the law is overall vague and developed with cryptocurrencies like Bitcoin in mind and not a more technologically sophisticated Monero. Indeed, privacy-oriented cryptocurrencies are addressed in a single point (pertaining to trading platform operations) in MiCA [3] that only stipulates deanonymization of trading parties be possible. This, however, leaves the door open for compliance, if only mechanisms for such deanonymization and effective policies for their use can be identified in Monero today or developed in the future. We try to predict what case law will evolve out of the regulations and conclude that some form of *transaction-level auditing* [9] is likely to be settled upon. In Sections 2.2 and 2.3, we then identify technical measures in the Monero protocol that can be expected to be used in implementing such auditing and map specific due diligence requirements onto them, thereby answering **RQ2**.[2] We do not explicate the exact framework for transaction-level auditing; instead, we show that no matter how it is implemented, it will fail to keep track of transactions published by third parties on behalf of the criminals, cautioning against

[1] https://www.getmonero.org/get-started/what-is-monero/ (accessed: 29 September 2025).

[2] Notably, we focus on Monero in its current form. Major updates, such as Seraphis and Jamtis, which have been work in progress for some years now but seem nowhere near completion, are out of scope.

relying on it for financial oversight. Indeed, we consider an ideal-functionality adversary that can inspect *all* traffic sent from the criminal's machine and has access to auditing instruments afforded by Monero. We then give a construction, in Sections 3 and 4, that enables the criminal's spending of moneroj[3] even in this extreme setting, thereby pointing to the futility of transaction-level auditing and partially answering **RQ3**. We finish by discussing a potential solution present in the Monero ecosystem, but conclude that, if applied at the required scale, it would substantially degrade Monero's privacy guarantees for *all* users, including non-EU residents. We conjecture that this cannot be avoided without substantial reworking of the Monero protocol, thus setting the direction of further Monero development. We devote most of the work to **RQ3** and, in answering it, develop what we call *anamorphic spending* that leverages *anamorphic signatures* [15] extended to the setting of ring signatures in order to thwart attempts at transaction-level auditing.

1.2 Notation

For the rest of the paper, let $\mathbb{G}$ denote the odd-order subgroup of the Edwards curve Ed25519. Let ℓ denote its (prime) order and G its distinguished generator. We write $u \cdot P$ to denote scalar multiplication of an elliptic curve point P by an integer u and $\mathbf{x}(P)$ to denote the x-coordinate of P. We adopt the notation from [14] and use $\mathcal{H}_p$ and $\mathcal{H}_n$ to denote cryptographic hash functions mapping arbitrary strings to elliptic curve points in $\mathbb{G}$ and scalars in $\mathbb{Z}_\ell$, respectively.

2 European Legal System and Monero

This section gives an overview of the legal framework regulating the trade of cryptocurrencies in the European Union (EU), which is our main focus in this work, and introduces the relevant Monero features along the way. For details regarding the Monero protocol, see [1, 14].

Before proceeding, note that the authors neither approve nor disapprove of the regulations in their current or future form. Instead, we aim to provide an account of the forseeable developments in the legislation and how it is or will be exercised, as well as provide a technical analysis of the capabilities and incapabilities of the underlying Monero system to comply with the law.

2.1 Markets in Crypto-Assets Regulation

The main legal act aiming to create a uniform and secure framework for the trade of *crypto-assets*[4] in the EU is Regulation 2023/1114 [3] on "markets in crypto-assets," or MiCA for short. We shall be concerned with *operating a trading*

[3] Units of the Monero currency.

[4] A crypto-asset is defined as "a digital representation of a value or a right that is able to be transferred and stored electronically using distributed ledger technology or similar technology" by [3]. In particular, this includes cryptocurrencies.

platform in the European Union (or wherever similar legislation may emerge) as defined in Article 3(1)(18) therein:

> *'operation of a trading platform for crypto-assets' means the management of ... systems, which bring together or facilitate the bringing together of multiple third-party purchasing and selling interests in crypto-assets ... in a way that results in a contract, either by exchanging crypto-assets for funds or by the exchange of crypto-assets for other crypto-assets.*

In particular, crypto *exchanges* fall into the category of trading platforms, even in the non-custodial setting, where they merely match asks and bids from an order book, and the actual trades get executed by the parties themselves without an intermediary. Running a trading platform is regulated in Article 76 of MiCA [3], which requires that service providers

> *lay down, maintain and implement clear and transparent operating rules for the trading platform ... [that at least] set the approval processes, including customer due diligence requirements commensurate to the money laundering or terrorist financing risk presented by the applicant in accordance with Directive (EU) 2015/849, that are applied before admitting crypto-assets to the trading platform.*

Here, MiCA [3] makes explicit reference to the "anti-money laundering" Directive 2015/849 [2], henceforth referred to simply as AML, giving legal grounds to obligating crypto trading platforms to exercise *due diligence* as defined by AML [2, Article 13(1)]:

> *Customer due diligence measures shall comprise:*
> *(a) identifying the customer and verifying the customer's identity ... obtained from a reliable and independent source;*
> *(b) identifying the beneficial owner and taking reasonable measures to verify that person's identity ...*
> *(c) assessing and, as appropriate, obtaining information on the purpose and intended nature of the business relationship;*
> *(d) conducting ongoing monitoring of the business relationship including scrutiny of transactions undertaken throughout the course of that relationship ...*

Möser et al. [20] note that, for Bitcoin, such customer due diligence could be exercised at the crypto-fiat boundary of the system, i.e., at an exchange platform, were it not for mixing services that obscure the transaction graph, preventing the identification of coin provenance (e.g., whether it is crime proceeds or not). In a privacy coin like Monero, however, the transaction graph is inherently secret (in a sense not much different from Bitcoin enhanced with mixing services), thereby making the utility of this kind of auditing questionable. At the same time, MiCA [3, Article 76(3)] reads:

> *The operating rules of the trading platform for crypto-assets shall prevent the admission to trading of crypto-assets that have an inbuilt anonymisation function unless the holders of those crypto-assets and their transaction history can be identified by the crypto-asset service providers operating a trading platform for crypto-assets.*

This suggests that the only hope for Monero's survival on the European market is effective technical measures for *transaction-level auditing* [9], where each trade involving Monero and at least one EU resident or legal person is inspected by the trading platform service provider (TPSP) according to AML's [2] due diligence rules. Indeed, many of the tools proposed in the literature for fighting cryptocurrency crime, such as tracing and blacklisting [19], cannot be applied to Monero unless transaction-level auditing is enforced. This answers **RQ1**.

In the following section, we identify existing mechanisms in Monero that could facilitate such oversight, assuming it to be a natural consequence of the existing legislation (and expecting it to soon be reflected either in case law or follow-up regulations). We do not give a concrete instantiation of an auditing framework, however; instead, we show that it cannot be effective no matter how it is implemented. For this, we show that even an all-powerful TPSP that inspects every single Monero transaction originating in the EU cannot be effective. We also remark that, in light of MiCA [3] being ambiguous about the TPSP's obligations, financial oversight institutions at member-state level could very well try to pursue a policy as stringent as this.

2.2 Anonymization Features of Monero

Monero uses a UTXO model, not unlike Bitcoin's, where each transaction points to heretofore unspent transaction outputs (UTXOs) as well as mints new UTXOs. A principal difference between Monero and, say, Bitcoin is that UTXOs are not directly addressed to *payment addresses*, which, in Monero, take the form $(K^v, K^s) \in \mathbb{G}^2$. Instead, a *one-time (stealth) address* $K^o \in \mathbb{G}$ is derived independently for each transaction output as

$$K^o = \mathcal{H}_n(r \cdot K^v) \cdot G + K^s ,$$

where r is chosen uniformly at random from $\mathbb{Z}_\ell^*$ and $R = r \cdot G$ is published in the transaction.[5] Therefore, $S = r \cdot K^v$ is a Diffie-Hellman key that the recipient can recompute given R and their private *view key* k^v such that $K^v = k^v \cdot G$. The secret S shared between the sender and the recipient is also used to encrypt the output amount a as well a Pedersen commitment trapdoor x, both of which are later required to spend the output (see Appendix A). Spend authority is proved by signing the transaction with the *output key* $k^o = \mathcal{H}_n(k^v \cdot R) + k^s$, where k^s is a private *spend key* such that $K^s = k^s \cdot G$. (Note that $K^o = k^o \cdot G$.) To hide the identity of the transaction's sender, a *ring signature* is used (see Appendix A) with $n-1$ *decoy* one-time addresses (transaction outputs) drawn at random from the Monero blockchain. For each transaction input, the sender is also required by the protocol to reveal a *key image* $\tilde{K} = k^o \cdot \mathcal{H}_p(K^o)$ and prove consistency thereof. Monero nodes keep track of the set of revealed key images to detect and prevent *double-spending*, i.e., referencing the same asset twice.

[5] We neglect the issue of diversification when multiple transaction outputs have the same recipient.

2.3 Transaction-Level Auditing in Monero

We now proceed to map the customer due diligence requirements defined by AML [2] and made applicable to, e.g., Monero exchanges by MiCA [3] onto the technical capabilities of Monero as it stands today, i.e., of Monero v0.18, thereby answering **RQ2**. The "scrutiny of transactions" from [2, Article 13(1)] is to be understood, in the broader context of the directive, as accounting for all transactions coming in to and going out from a user's permanent address (K^v, K^s), including their exact amounts.[6] True beneficiaries of outgoing transactions must also be identified. In the following paragraphs, we study these obligations one by one.

As argued in Section 2.1, it is natural to consider an edge case where an all-powerful TPSP acts as a watchdog, sitting between its clients and the public ledger, and inspects every transaction broadcast by or addressed to an EU citizen.[7] This could be seen as an ideal-functionality model for transaction-level auditing. We note that data collected this way (in compliance with AML) would be withheld from the public, thereby preserving anonymity in the eyes of other Monero network participants and making the policy seem a viable compromise between accountability and privacy. We must caution against relying on transaction-level auditing for MiCA [3] compliance, however. Indeed, below, we briefly identify mechanisms existing in Monero today that are natural candidates for implementing transaction-level auditing in such centralized, i.e., strongest possible, form, but follow this with a novel application of anamorphic cryptography that makes *all* transaction-level auditing attempts fail to detect transactions going out from the EU.

We keep the setting non-custodial assuming that (at least) the private spend key k^s remains solely with the client (and not the trading platform). Indeed, the directive [3] speaks of "ownership rights," and there exists a legal precedent in Europe to equate the owner of tokens to the "person who has the right to dispose of" them [6]. Importantly, however, we may consider, instead of a client and their trading platform service provider, an exchange subjected to a continuous audit by an external authority. Hiding the very fact of transacting or even just the truth about transaction amounts could allow such an exchange to, e.g., embezzle the funds escrowed with it without immediately revealing its insolvency. All reasoning that follows in Sections 3 and 4 applies just the same to this case.

Finally, whenever we talk about identification of transaction's participants in this work, we are only concerned with finding their Monero addresses (K^v, K^s) and assume these can then be mapped to real-world identities by the authorities, e.g., via methods and heuristics developed for Bitcoin for which there is extensive literature available [5, 17, 23].

Incoming Transactions with Amounts. A TPSP is obligated to analyze the transaction history of a client, hence, in particular, identify incoming assets.

[6] Note that the platform service provider may be obligated by the law to disclose, e.g., to a revenue service, all transactions whose amounts exceed a certain threshold.

[7] In practice, the TPSP could block the transaction if not convinced of its legitimacy.

This can be achieved by disclosing the private view key k^v to the TPSP granting them insight into transactions sent to the client's payment address. The private view key also allows for the reconstruction of Diffie-Hellman keys $S = k^v \cdot R$ and the decryption of transaction amounts. We expect that, in light of MiCA [3], trading platforms will have a valid legal reason to request private view keys of their clients. Compare this with prevailing guidance of financial authorities in Switzerland on money laundering [25] dictating "suitable technical means" be used to prove ownership of a wallet. It is natural, in the case of Monero, to extend any such provision to require disclosure of k^v as well.

Outgoing Transactions with Amounts and True Beneficiaries. According to AML [2], a TPSP is obligated to identify the party originating a transaction as well as monitor the business relationship of the transaction participants. This also involves identification of the exact transaction amount and its true beneficiary. Identification of the amount is naturally achieved by disclosing to the TPSP the Diffie-Hellman key S that the sender computes as $r \cdot K^v$ for the recipient's view key K^v. A straightforward proof of equality of discrete logarithms can then be used to convince the TPSP that S corresponds to $R = r \cdot G$ (also included in the transaction) and K^v. Let $\mathsf{EqLog}_{G_1,\ldots,G_d}(K_1,\ldots,K_d)$ denote a non-interactive proof of equality of discrete logarithms of $K_1,\ldots,K_d$ to bases $G_1,\ldots,G_d$, respectively.[8] The sender can convince the TPSP that an output is addressed to (K^v, K^s) by presenting them with (S, π), where $\pi = \mathsf{EqLog}_{G,K^v}(R, S)$. Given this and the one-time address K^o of the output, the TPSP can check that $K^s = K^o - \mathcal{H}_n(S) \cdot G$ and verify π. If both checks pass, the TPSP is convinced. This construction is referred to as an `OutProof` in [14]. Knowledge of S also immediately enables the TPSP to decrypt the transaction (output) amount.

One problem that remains to be addressed is that a user may hand the TPSP transactions originated by someone else. Since Monero uses ring signatures to authorize transactions, the TPSP's seeing in the ring a one-time address K^o corresponding to the client's permanent address (K^v, K^s) *does not* convince them that it is the client who is actually making a transfer. Any input in the ring could actually be getting spent in this transaction. The client should prove that the key image $\tilde{K}$ published in the transaction corresponds to K^o, i.e., that $\tilde{K} = k^o \cdot \mathcal{H}_p(K^o)$. This can be done by sending $\mathsf{EqLog}_{G,\mathcal{H}_p(K^o)}(K^o, \tilde{K})$ to the TPSP.[9]

2.4 Pitfalls and Shortcomings

Having established the due diligence requirements following from [2, 3], we will proceed to study the pitfalls awaiting implementers who turn to transaction-level auditing. Clearly, out-of-band channels, bypassing the TPSP, e.g., by means of

[8] A description of such a proof can be found in [14, Section 3.1].

[9] A more complex proof for this is referred to as an `UnspentProof` in [14].

a VPN, enable the evasion of transaction-level scrutiny. This could, in principle, be outlawed, but enforcing such a ban would be non-trivial. We, however, choose to focus on a more fundamental problem, namely, that, even if a robust framework for transaction-level auditing is established and implements the above measures, it is *still* not enough to prevent money laundering or otherwise illegal transfers of moneroj (XMR). This result relies on the notion of *anamorphic cryptography* [15,21] (summarized in Appendix B) and points to the futility of many regulatory efforts. Specifically, however the transaction-level auditing framework is implemented, even if every transaction broadcast by an EU citizen goes through the hands of the auditor (TPSP), then, still, in-band channels are available for criminals to send XMR offshore in a provably undetectable way.

Before proceeding, observe one final discrepancy between the legislation and the available technology. In a privacy coin like Monero, the full provenance of funds can never be established, even if all EU residents are subjected to transaction-level auditing. This is due to transacting with counterparties not based in the EU. European platforms would have no legal grounds (nor technical capabilities) to request view keys or `OutProof`s from foreign actors and so identifying incoming assets as, e.g., crime proceeds would remain impossible, all the while EU residents could not be held accountable for unsolicited reception of funds (note that Monero does not support any notion of a "refund"). This clashes with the intent of the lawmakers expressed in recital 77 of MiCA's [3] preamble:

> *In order to ensure the continued protection of the financial system of the Union against the risks of money laundering and terrorist financing, it is necessary to ensure that crypto-asset service providers carry out increased checks on financial operations involving customers and financial institutions from third countries listed as high-risk third countries . . .*

Indeed, "increased checks" on operations involving transfer of Monero from third-country counterparties to the EU are hardly viable without international cooperation.

3 Anamorphic Channel in CLSAG Signatures

In this section, we construct an *anamorphic channel* [15,21] within the Monero protocol that shall enable in-band transfer of spending rights from an EU citizen, Alice, to an offshore accomplice, Bob, all without the TPSP's realizing, even if every single transaction broadcast by Alice (and, indeed, every byte that leaves her machine) goes through the TPSP's hands, and the TPSP implements the auditing mechanisms described above. To start, we give a construction that enables Alice, in possession of a *sender double key* `sdk`, to embed covert messages in Monero transactions in such a way that the *anamorphic transactions* carrying the secret messages are indistinguishable from regular Monero traffic even to a TPSP in possession of Alice's view key k^v and requiring `OutProof`s [14] for all transactions she broadcasts. In fact, the TPSP may also have possession of the double key sdk (the existence and knowledge of which Alice can plausibly deny and should deny). Only Bob (who need not coincide with any of the anamorphic transaction's recipients) with a matching *receiver double key* rdk will be able to recover the covert messages.

Hybrid Encryption on Chain. For each input to a Monero transaction, a CLSAG ring signature [11] is produced (see Appendix A), each using $n > 1$ public keys that form a *ring*, with the signer's key at position π and $n - 1$ *decoys*. Let $r_1, \ldots, r_n \in \mathbb{Z}_\ell$ denote the n nonces published as part of a CLSAG signature σ. Importantly, each r_i for $i \neq \pi$ is supposed to be drawn uniformly at random from $\mathbb{Z}_\ell$, while r_π is derived deterministically and depends on the other nonces.[10] Instead of randomly choosing the decoy r_is, we are going to use them for covert communication. This idea has been explored already by Alsalami and Zhang in [4], but instead of adopting their construction (designed for a signature scheme from an older version of Monero), we propose our own, which works in an asymmetric setting, enabling multiple mutually distrustful senders to communicate with the receiver holding rdk. We also enhance [4] by using a stream cipher-like construction for encryption, as opposed to a block cipher, making the relevant security reductions more immediate. Our construction uses one of the nonces r_i to run a Diffie-Hellman key exchange and the remaining $n-2$ random r_is to hide ciphertexts. Given that, in practice, $n = 16$,[11] this results in a covert channel of bandwidth around 440 bytes per CLSAG signature, i.e., per one input to a transaction.

Let $V = m \| T$ denote the plaintext to be hidden in a CLSAG signature σ, where m is the message and T is the tag of a secure message authentication code computed with a key ξ that we shall establish momentarily to be shared between the sender and the receiver. Write $\mathsf{Verify}(m, T, \xi)$ to denote verifying this tag. The covert receiver will not know π, the position of the non-random nonce, a priori and thus will have to make up to n trial decryptions for different guesses, using MAC verification to recognize the correct one. Encode V as an integer in base ℓ with at most $n - 2$ digits:

$$V = (v_{n-2} v_{n-3} \cdots v_1)_\ell = \textstyle\sum_{i=1}^{n-2} \ell^{i-1} v_i \,.$$

Generate *keystream* $u_1, \ldots, u_{n-2} \in \mathbb{Z}_\ell$ using a PRNG seeded with the shared key ξ and let

$$e_i = v_i + u_i \pmod{\ell} \quad \text{for } 1 \leq i \leq n-2 \,. \tag{1}$$

The values e_i are indistinguishable from uniformly random on $\mathbb{Z}_\ell$. We shall use an extra value e_0 to run an ephemeral-static Diffie-Hellman protocol with

[10] The reference implementation chooses the nonces from the multiplicative subgroup $\mathbb{Z}_\ell^*$ only, rejecting $r_i = 0$, despite [11,14] saying otherwise. See `random32_unbiased` at https://github.com/monero-project/monero/blob/b591866fcfed400bc89631686655aa769ec5f2dd/src/crypto/crypto.cpp/#L123 (accessed: 29 September 2025). This adds an extra step to the proof of Theorem 1 but otherwise has no bearing on the validity of the claim itself.

[11] See, e.g., https://xmrchain.net (accessed: 29 September 2025).

the sender double key sdk and set ξ to be the output of this protocol. Finally, we shall set the CLSAG nonces r_i to the ciphertexts e_j skipping r_ℓ which is produced deterministically as per CLSAG specification [11]. If the covert receiver knew π, they could correctly recover $e_0, \ldots, e_{n-2}$ (by excluding r_π) and, given rdk, complete the Diffie-Hellman protocol started in e_0, compute ξ, and recover the keystream $u_1, \ldots, u_{n-2}$. This would enable them to undo Equation (1) and recover the plaintext V. Without a priori knowledge of π, the covert receiver can instead recover "trial" plaintexts $V' = (v'_{n-2} v'_{n-3} \cdots v'_1)_\ell$, for different guesses of π, and, for each of them, parse V' as $m' \| T'$ and check if $\mathsf{Verify}(m, T, \xi) = 1$. (Note that, this way, the covert receiver, in possession of rdk, finds π, thus breaking the anonymity of the transaction sender.) We now proceed to give a construction for running the Diffie-Hellman protocol covertly in e_0, so that e_0 looks (unconditionally) uniformly random.

Random-Looking ECDH in CLSAG. We shall generalize Möller's construction for hybrid encryption with pseudorandom ciphertexts [18] to non-binary elliptic curves. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve in short Weierstrass form over the field $\mathbb{Z}_\ell$,[12] and let $d \in \mathbb{Z}_\ell^*$ be a non-square. The (quadratic) *twist* of E by d is the curve $E_d : dy^2 = x^3 + ax + b$. Let the covert receiver, call him Bob, choose $a, b, d \in \mathbb{Z}_\ell^*$, with d non-square so that the curve E *and* its quadratic twist E_d are both secure. In particular, both orders $|E|$ and $|E_d|$ should be prime, and the decisional Diffie-Hellman assumption should hold for both groups. Note that the former implies that every $x \in \mathbb{Z}_\ell$ corresponds either to a pair of distinct points on E (if x^3+ax+b is a square in $\mathbb{Z}_\ell$) or a pair of points on E_d (otherwise). Indeed, points of the form $(x, 0)$ would have order 2, contradicting the primality of the order of the whole group. We give a construction (following [18]) that produces points (with known discrete logarithms) on either E or E_d in such a way that their x-coordinates are perfectly indistinguishable from values drawn uniformly at random from $\mathbb{Z}_\ell$.[13] Let $\eta = |E| - 1$ denote the number of finite (excluding the point at infinity ∞) points on E, and, similarly, let $\eta_d = |E_d| - 1$ denote the number of finite points on E_d. It can be shown that $\eta + \eta_d = 2\ell$ (see, e.g., [10]). Let Bob fix $B \in E$ and $B_d \in E_d$ and generate a *double key pair* by drawing uniformly at random scalars $s \in \{1, \ldots, \eta\}$ and $s_d \in \{1, \ldots, \eta_d\}$ and computing $Y = s \cdot B$ and $Y_d = s_d \cdot B_d$. Let the sender double key be

$$\mathsf{sdk} = (Y, Y_d, a, b, d, \eta, \eta_d, B, B_d)\,,$$

and the receiver double key

$$\mathsf{rdk} = (s, s_d, a, b, d, \eta, \eta_d, B, B_d)\,.$$

As the proof of Theorem 1 will show, sdk can be published either off chain or on chain in an appropriately tagged transaction for both the sender, call her Alice,

[12] Itself a scalar field of another elliptic curve group, namely, $\mathbb{G}$.

[13] Choosing points on just one curve, say, E, would not work, since the values of x for which $x^3 + ax + b$ is not a square in $\mathbb{Z}_\ell$ would occur with probability 0.

and the TPSP to read, since knowledge of sdk will not help the TPSP convict Alice. Indeed, the covert ciphertext is indistinguishable from a random string.[14] To send a covert message to Bob, let Alice toss an asymmetric coin to select one of the curves, E or E_d. Specifically, let her choose E with probability $\eta/(\eta+\eta_d)$ and E_d with probability $\eta_d/(\eta+\eta_d)$. Let $\tilde{E} \in \{E, E_d\}$ denote the curve chosen for the transfer, $\tilde{B} \in \{B, B_d\}$ the corresponding basepoint of order $\tilde{\eta}+1$, with $\tilde{\eta} \in \{\eta, \eta_d\}$, and $\tilde{Y} \in \{Y, Y_d\}$ the corresponding component of sdk. Let Alice choose uniformly at random a scalar $q \in \{1, \dots, \tilde{\eta}\}$ and compute an ephemeral public key $Q = q \cdot \tilde{B}$ as well as the corresponding shared secret $S = q \cdot \tilde{Y}$. Finally, let Alice set $e_0 = \mathbf{x}(Q)$ and use $\xi = \mathcal{H}_n\big(\mathbf{x}(S)\big)$ for the encryption (and authentication) of the anamorphic message V as described earlier.

For each transaction published in the ledger, Bob can check the accompanying CLSAG signatures for hidden ciphertexts. Specifically, for each signature σ, Bob can recover $e_0, \dots, e_{n-2}$ (there are n possibilities for r_π which must be excluded from σ to obtain this sequence). Given e_0, Bob can check if $e_0^3 + ae_0 + b$ is a square modulo ℓ. If so, Bob should assume $e_0 = \mathbf{x}(Q')$ for $Q' \in E$ and compute $\xi' = \mathcal{H}_n\big(\mathbf{x}(s \cdot Q')\big)$. Otherwise, Bob should assume Q' belongs to E_d and compute $\xi' = \mathcal{H}_n\big(\mathbf{x}(s_d \cdot Q')\big)$. In either case, Bob can seed a PRNG with ξ', generate the keystream $u_1, \dots, u_{n-2}$, and solve Equation (1) for a trial decryption $V' = m' \| T'$. If $\mathsf{Verify}(m', T', \xi') = 0$ for all guesses of π, then σ did not contain a hidden ciphertext. Otherwise, Bob has recovered m.

In the theorem below, we extend the notion of an anamorphic signature scheme due to Kutyłowski et al. [15] (elaborated in Appendix B), formalized in terms of a distinguishing game with a *dictator* in possession of the signing key, to ring signatures in a natural way, i.e., we may, without loss of generality, assume the dictator chooses the decoy public keys. We also let the security parameter be $\lambda = |\ell|$.

Theorem 1. *CLSAG [11] is an anamorphic (ring) signature scheme (see Definition 2) in the random oracle model with the associated anamorphic triplet (see Definition 1) given by the construction above.*

Before proceeding with the proof, two technical lemmata are needed.

Lemma 1. *Let $\tilde{E}$ be the elliptic curve sampled as above, (B, B_d, Y, Y_d) the keys sampled as above, and q uniform in $\{1, \dots, \tilde{\eta}\}$. The distribution of the random variable $X_0 = (q \cdot \tilde{B}, q \cdot \tilde{Y}, \tilde{Y}, \tilde{B})$ is indistinguishable from $X_1 = (q \cdot \tilde{B}, w \cdot \tilde{B}, \tilde{Y}, \tilde{B})$ for w uniform in $\{1, \dots, \tilde{\eta}\}$.*

Proof. Let $\mathcal{A}$ be an efficient (i.e., probabilistic polynomial-time) distinguisher. Write $X_i^{\mathcal{E}}$ for the random variable distributed according to the conditional probability of X_i conditioned on the event $\tilde{E} = \mathcal{E}$. Observe that, for any $i \in \{0, 1\}$,

$$\Pr[\mathcal{A}(X_i) = 1] = \frac{\eta}{\eta + \eta_d} \cdot \Pr[\mathcal{A}(X_i^E) = 1] + \frac{\eta_d}{\eta + \eta_d} \cdot \Pr[\mathcal{A}(X_i^{E_d}) = 1] \,.$$

[14] Note how this is a stronger notion than mere *key-privacy* of the underlying hybrid encryption scheme.

Write

$$\mathbf{Adv}(\mathcal{A}, U, W) = |\Pr[\mathcal{A}(U) = 1] - \Pr[\mathcal{A}(W) = 1]|$$

for the *advantage* of $\mathcal{A}$ in distinguishing the distributions U and W. It follows (by triangle inequality) that

$$\mathbf{Adv}(\mathcal{A}, X_0, X_1) \leq \frac{\eta}{\eta + \eta_d} \cdot \mathbf{Adv}(\mathcal{A}, X_0^E, X_1^E) + \frac{\eta_d}{\eta + \eta_d} \cdot \mathbf{Adv}(\mathcal{A}, X_0^{E_d}, X_1^{E_d}),$$

but the problem of distinguishing X_0^E from X_1^E is exactly the decisional Diffie-Hellman problem on the curve E. Similarly, distinguishing $X_0^{E_d}$ from $X_1^{E_d}$ is the decisional Diffie-Hellman problem on the curve E_d. By assumption about E and E_d, there exist negligible functions $\varepsilon_{\mathsf{ddh}}^E$ and $\varepsilon_{\mathsf{ddh}}^{E_d}$ such that

$$\mathbf{Adv}(\mathcal{A}, X_0^E, X_1^E) \leq \varepsilon_{\mathsf{ddh}}^E \quad \text{and} \quad \mathbf{Adv}(\mathcal{A}, X_0^{E_d}, X_1^{E_d}) \leq \varepsilon_{\mathsf{ddh}}^{E_d}.$$

Set $\varepsilon = \max\{\varepsilon_{\mathsf{ddh}}^E, \varepsilon_{\mathsf{ddh}}^{E_d}\}$ and note that a (point-wise) maximum of two negligible functions is itself negligible. We have

$$\mathbf{Adv}(\mathcal{A}, X_0, X_1) \leq \frac{\eta}{\eta + \eta_d} \cdot \varepsilon + \frac{\eta_d}{\eta + \eta_d} \cdot \varepsilon = \varepsilon,$$

which completes the proof. □

Lemma 2. *Sample $\tilde{E}$ as above and Q uniformly in $\tilde{E} \setminus \{\infty\}$. Then $\mathbf{x}(Q)$ is uniformly random in $\mathbb{Z}_\ell$.*

Proof. Recall that $\eta + \eta_d = 2\ell$ (cf. [10]). It follows that, for each $x \in \mathbb{Z}_\ell$, the "right" curve is chosen with probability $\tilde{\eta}/2\ell$, and, out of the $\tilde{\eta}$ finite points $Q \in \tilde{E}$, each equally likely, two have $\mathbf{x}(Q) = x$. □

Proof (of Theorem 1). Assume without loss of generality that $\pi = n$. Write $\mathsf{G}_0 = \mathsf{AnamorphicG}$ for the *anamorphic game* (see Definition 2), where an efficient dictator $\mathcal{D}$, in possession of the CLSAG signing key,[15] interacts with an oracle implementing the above construction. Write p_i for the probability of $\mathcal{D}$ returning 1 in game G_i and $t = \mathsf{poly}(\lambda)$ for the (upper bound on the) number of queries $\mathcal{D}$ makes to the oracle.

Let G_1 be exactly like G_0 except the oracle seeds the PRNG with $\mathbf{x}(R)$ for R sampled uniformly from $\tilde{E} \setminus \{\infty\}$ as opposed to with $\mathbf{x}(q \cdot \tilde{Y})$. The dictator's advantage in deciding the game based on a single oracle query is bounded by a negligible function ε according to Lemma 1. When making $t(\lambda)$ queries, the advantage is instead bounded by $t(\lambda) \cdot \varepsilon$, since the distributions involved are polynomial-time constructible.[16] It follows that

$$|p_0 - p_1| \leq t(\lambda) \cdot \varepsilon.$$

[15] As we shall see, we may also assume the dictator knows the sender double key sdk.

[16] A standard reduction from a single-sample distinguisher to a multi-sample distinguisher involves choosing at random an index i between 1 and $t(\lambda)$ and getting the first $i-1$ samples from the first distribution, the last $t(\lambda) - i$ samples from the second distribution, and setting the ith sample to the reduction's input. The reduction's advantage is then $1/t(\lambda)$th that of the multi-sample distinguisher.

Note that Lemma 1 relies on the decisional Diffie-Hellman assumption and, crucially, *not* on the secrecy of $\tilde{Y}$, which shows that the result holds even in the case the dictator knows sdk, thereby allowing Bob to publish it in the Monero ledger for everyone to see. In G_2, let the oracle seed the PRNG with a uniformly random $\xi \in \mathbb{Z}_\ell$ as opposed to $\mathcal{H}_n\big(\mathbf{x}(R)\big)$. In the random oracle model, the advantage in distinguishing between games G_1 and G_2 is bounded by the probability of querying the random oracle $\mathcal{H}_n$ on $\mathbf{x}(R)$ for R uniform in $\tilde{E}$.[17] Write $\mathbf{x}(\tilde{E})$ for the set of x-coordinate corresponding to points on $\tilde{E}$ and assume without loss of generality that $|E| \leq |E_d|$. By the Hasse bound:

$$|\eta - \ell| \leq 2\sqrt{\ell}$$

and since $\ell \geq 2^{\lambda-1}$, we have that

$$|p_1 - p_2| \leq \frac{t(\lambda)}{|\mathbf{x}(\tilde{E})|} \leq \frac{t(\lambda)}{|\mathbf{x}(E)|} = \frac{2 \cdot t(\lambda)}{\eta} \leq t(\lambda) \cdot 2^{-\lambda+3} \,.$$

Let G_3 differ from G_2, in that the oracle sets e_0 to be uniformly random in $\mathbb{Z}_\ell$ as opposed to $\mathbf{x}(Q)$. By Lemma 2, we have $p_2 = p_3$.[18] Let G_4 differ from G_3, in that the oracle chooses the keystream $u_1, \ldots, u_{n-2}$ uniformly at random from $\mathbb{Z}_\ell$. Then,

$$|p_3 - p_4| \leq \varepsilon_{\mathsf{prng}} \,,$$

where $\varepsilon_{\mathsf{prng}}$ is the maximum advantage an efficient adversary may have in distinguishing the PRNG output[19] from a uniformly random string. Then, in G_5, let the oracle choose the *nonces* $e_1, \ldots, e_{n-2}$ uniformly at random from $\mathbb{Z}_\ell$ instead of the keystream. Clearly, this is only a conceptual change, and $p_4 = p_5$. Observe that $\mathsf{G}_5 = \mathsf{RealG}$ corresponds exactly to the *real game* (see Definition 2), in which $\mathcal{D}$ interacts with an honest CLSAG signer, and

$$|p_0 - p_5| \leq t(\lambda) \cdot \varepsilon + t(\lambda) \cdot 2^{-\lambda+3} + \varepsilon_{\mathsf{prng}} \,,$$

which is negligible. □

4 Anamorphic Spending of XMR

We now show how the channel established above can be leveraged to transfer spending rights from Alice, subject to the TPSP's transaction-level auditing, to her offshore accomplices, who operate outside the EU's jurisdiction, i.e., have unencumbered access to the Monero network and ledger. Suppose direct transfer would be denied as violating AML control rules for transactions leaving the

[17] Note that the dictator learns $\tilde{E}$ based on e_0.

[18] A randomly chosen scalar q induces a uniformly random permutation of the prime-order group of points on $\tilde{E}$; hence, Q *is* uniformly random on $\tilde{E}$.

[19] Of length $\lambda \cdot n \cdot t(\lambda)$.

EU. We will have Alice broadcast *anamorphic transactions* that cannot be distinguished from regular transactions by the TPSP but contain tokens, most importantly the output key k^o, required to spend some other UTXO of Alice's.

Consider a TPSP provisioned with the private view key k^v of every user that registers with the platform, including Alice. Furthermore, assume the extreme (centralized) case, where every transaction broadcast by Alice passes through the hands of this TPSP who has the right to request `OutProofs` [14] as described in Section 2.3 to identify the recipients of outgoing transactions.[20] We introduce the notion of *anamorphic spending* where Alice transfers the spending rights for a UTXO to her accomplice, Bob, via an anamorphic channel established in Section 3. The corresponding funds will, from the point of view of the TPSP engaging in transaction-level auditing, seem to have never been spent in the EU but will nonetheless surface on a foreign market. Let a, x, and k^o be the amount, the output commitment trapdoor (see Appendix A), and the output key, respectively, of Alice's UTXO τ. Alice can create *any* transaction T, unrelated to τ, that sends some other UTXO to *anyone*, and embed a, x, and k^o in an anamorphic channel to Bob hidden within said transaction. Specifically, when signing (some input of) T, Alice, using Bob's double key `sdk`, follows the procedure laid out in Section 3 to produce an anamorphic signature (thus making T into an anamorphic transaction) that cannot be distinguished from a signature generated honestly according to the CLSAG specification (indeed, the TPSP here is subsumed by the even more capable dictator of Theorem 1) and therefore does not raise the suspicion of the TPSP (Alice is granted deniability).

Bob can now scan the Monero ledger, looking for transactions that carry anamorphic messages to him. Once Bob recognizes T, he can recover a, x, and k^o, and then be able to spend τ.[21] Since his access to the ledger is unencumbered by any trading platform's due diligence obligations, he can broadcast his transaction T' that takes τ as input. Anonymity mechanisms of Monero then work against the TPSP in Europe that cannot tell if τ was used as a decoy (recall Section 2.2) or actually spent in T'. Importantly, Bob does not learn Alice's long-term keys (k^v, k^s) and instead receives a token, namely, k^o, that only authorizes him to spend τ and no other UTXO. Note how this enables Alice to funnel XMR outside the EU without compromising her account. The money funneled offshore can later return to the EU using regular Monero channels (recall from Section 2.4 that such channels are hard to regulate and any reception of funds could be plausibly argued unsolicited), thus affording a money-laundering scheme. The Monero ledger serves here as a perfect anonymous communication channel: at no point does Alice communicate directly with Bob, except via the *bulletin board* afforded by the ledger, thwarting any wiretapping attempts by the authorities.

We note that it may be custodial crypto exchange services that engage in anamorphic spending and thus, e.g., successfully hide their insolvency from both their clients and the auditors equipped with view keys k^v to the exchange's

[20] The TPSP may also request $\mathsf{EqLog}_{G,\mathcal{H}_p(K^o)}(K^o, \tilde{K})$ (recall Section 2.3) to ensure the transactions Alice publishes actually spend Alice's funds.

[21] Bob can easily recognize τ by its one-time address $K^o = k^o \cdot G$.

accounts on the ledger. Also, given the trend to increase the ring size n, the anamorphic channel of Section 3 gets wider over time, enabling making many subliminal transfers, each possibly to a different recipient, all using a single anamorphic transaction, intuitively reducing the effectiveness of heuristics in detecting anamorphic spending.

In light of the above, we arrive at the following conclusion:

Conclusion 1. *Neither auditing at the crypto-fiat boundary nor transaction-level auditing is effective in the case of Monero. The former offers little insight into transaction history due to in-built anonymity features of the currency, and the latter remains susceptible to anamorphic spending with non-EU counterparties (which are given extra emphasis in the preamble of [3]; see, e.g., recital 77 therein), even if all out-of-band channels are shut.*

Note that we have assumed a model of the TPSP that is easily argued compliant with AML's [2] due diligence requirements and yet fails to account for the full scope of technical capabilities at the financial criminals' disposal, anamorphic cryptography included. This highlights a danger in implementing the regulations [2, 3] carelessly (or maliciously).

4.1 Defecting Bob

In the above construction, we, importantly, assume that Bob remains "loyal" to Alice and does not collaborate with any of the trading platforms that know Alice's private view key k^v. Indeed, if Bob were to disclose k^o obtained from the anamorphic transaction to the platform's provider, they would together compromise Alice's account. This is because a party in possession of both an output key k^o (for a transaction with public key R) and its corresponding private view key k^v immediately learns the spend key $k^s = k^o - \mathcal{H}_n(k^v \cdot R)$, granting them spend authority over *all* Alice's UTXOs.[22] If a law enforcement agency in pursuit of Alice obtained k^s, they could effectively "freeze" her assets. To not run the risk of Bob turning rogue and "ratting" on her, Alice could use a straw man (a "mule"), Charlie. For a "cut," Charlie would forward to Bob, via the anamorphic channel, the output keys (together with a and x) for transactions coming, via the public ledger, from Alice to Charlie. Now, only Charlie's account in Monero would be at risk. A criminal organization could control multiple money mules like Charlie to further conceal its activities. Furthermore, if Charlie sells both his private keys to Alice, he need not ever again engage with the Monero network and can serve only as a front registered with the trading platform. Alice can make anamorphic transactions T addressed to Charlie but secretly (through the anamorphic channel) authorize Bob to spend their outputs τ as well. Assuming some "honor among thieves," Charlie will not spend them before Bob does.

Interestingly, note that it is not clear who the legal owner of the coins is (and who is responsible for further activities involving them) after the execution of

[22] This is not prevented even by the use of *subaddresses*, which we chose not to introduce in this paper for brevity.

the anamorphic spending protocol as above. Equating the owner to the person technically capable of spending the funds (see [6] for a precedent) would suggest that Alice, Bob, and Charlie have joint ownership of the asset at this point. MiCA [3] takes the notion of crypto-asset ownership and custody for granted and gives the technical nuances no treatment.

4.2 Air-Tight Auditing and Consequences

We now revisit **RQ3** and briefly discuss the capacity of Monero (in its current form) to accommodate a more robust auditing framework. One way to satisfy the directive's [2] requirements would be for TPSPs to compel their clients to disclose key images $\tilde{K}$ of all their assets, on top of employing the tools of Section 2.3.[23] For honest users, this would not violate privacy any more than transaction-level auditing where the TPSP already knows when each UTXO gets spent. It would, however, thwart anamorphic spending as described above. Indeed, the auditor with knowledge of the key images of a user's assets can tell whenever any one of them gets spent, no matter who broadcasts the transaction or in what jurisdiction. This is because the corresponding key image *must* be revealed on chain when spending a UTXO. We highlight here the alignment between the law and the technical requirements of the Monero protocol: key images' correspondence to spends is integral to double-spending prevention, therefore it is enforced by the protocol itself.

There are, however, two problems with this approach. First, it encroaches on the privacy of non-EU residents in that their anonymity guarantees are reduced, since collaborating EU-based TPSPs could rule out decoy keys of EU residents of which they know the key images. Second, there is no way to enforce the policy in a non-custodial setting, because a separate proof protocol (showing the consistency of the disclosed key image) must be played out between the user's wallet and the platform for each incoming asset. As long as the wallet remains offline, the account may accrue assets without revealing the key images, and the user cannot be held accountable for *not* using their wallet (e.g., keeping it "cold"). If Alice downloads her private keys, she may derive the output keys and transfer them anamorphically to Bob using an accomplice's, Charlie's, account.

Future of Monero. There have been numerous proposals for future directions of Monero, including empowering view-only wallets by changing the way key images are computed. This could, incidentally, grant auditors or TPSPs offline insight into outgoing transactions of their subjects or clients. There is, however, no official schedule as to when, if ever,[24] such changes may be made to Monero.

[23] This is a realistic concern considered by Monero developers, see Justin Berman's talk at MoneroKon 2022: https://www.youtube.com/watch?v=xGEBRQU1lzw (accessed: 29 September 2025).

[24] See https://gist.github.com/tevador/50160d160d24cfc6c52ae02eb3d17024?permalink_comment_id=5330787#gistcomment-5330787 (accessed: 29 September 2025).

Likewise, *full chain membership proofs*, that would extend the anonymity set to *all* UTXOs in the Monero ledger, are also on the development roadmap.[25]

5 Related Work

The study of establishing covert channels goes back to Simmons [24], with public blockchains only recently identified as perfect platforms for realizing them [4, 7, 12, 16, 26, 27]. The notion of anamorphic cryptography is introduced in [21], with [15] further introducing anamorphic signatures and [22] introducing public-key anamorphic cryptography. Tiemann et al. [26] study using the Bitcoin blockchain as a secure chat (bulletin board). Their construction, however, requires the transaction creator to reveal to the receiver the private keys used to sign the transactions. Guo et al. [12] present a covert channel in Monero transactions with capacity of $n-1$ *bits* per transaction input, i.e., approximately $\lg(\ell) = 253$ *times* smaller than ours. In [16], the authors propose a covert channel in Monero transactions using the transaction amount as carrier. Alsalami and Zhang [4] study Monero in the context of covert broadcast and propose embedding ciphertexts in Monero's (old version of) ring signatures. Our construction in Section 3 can be viewed as an extension of their work that does away with reliance on pre-shared symmetric keys. Keller et al. [13] show a scheme for proving a UTXO was only used as a decoy. It amounts to revealing the key image and proving its consistency.

6 Conclusions

We have highlighted the discordance between the emerging European law and the technological reality of trading Monero. In particular, we have considered the MiCA regulation [3] and argued that, in order to comply with Article 76(3) therein, some form of transaction-level auditing must be implemented to account for transaction histories of EU residents. We then considered the perfect functionality of an auditor that acts as a watchdog through the hands of which every transaction must pass and precluded the use of any out-of-band channels. In this extreme setting, we have shown that robust financial oversight remains elusive, despite on-paper compliance with the regulation [3] and the AML directive [2], due to a novel notion of anamorphic spending, which applies anamorphic signatures [15] to the blockchain setting. Anamorphic Monero transactions, indistinguishable from regular ones, can be leveraged to continue running money-laundering schemes with offshore accomplices largely unhindered, despite the Union's attempts at oversight.

[25] See https://www.getmonero.org/resources/roadmap/ (accessed: 29 September 2025).

References

1. Monero Inflation Checker. https://www.moneroinflation.com/, accessed: 29 September 2025
2. Directive (EU) 2015/849 of the European Parliament and of the Council of 20 May 2015 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, amending Regulation (EU) No 648/2012 of the European Parliament and of the Council, and repealing Directive 2005/60/EC of the European Parliament and of the Council and Commission Directive 2006/70/EC. Official Journal of the European Union **L 141**, 73–117 (2015)
3. 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. Official Journal of the European Union **L 150**, 40–205 (2023)
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
5. 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)
6. Benson, V., Adamyk, B., Chinnaswamy, A., Adamyk, O.: Harmonising cryptocurrency regulation in Europe: Opportunities for preventing illicit transactions. European Journal of Law and Economics **57**(1-2), 37–61 (2024). https://doi.org/10.1007/s10657-024-09797-w
7. 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
8. Chainanalysis: The 2024 crypto crime report (2024), https://go.chainalysis.com/crypto-crime-2024.html
9. 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
10. Fouque, P., Lercier, R., Réal, D., Valette, F.: Fault attack on elliptic curve Montgomery ladder implementation. In: Breveglieri, L., Gueron, S., Koren, I., Naccache, D., Seifert, J. (eds.) Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008. pp. 92–98. IEEE Computer Society (2008). https://doi.org/10.1109/FDTC.2008.15
11. Goodell, B., Noether, S., Blue, A.: Concise linkable ring signatures and forgery against adversarial keys (2020), https://api.semanticscholar.org/CorpusID:215542434
12. Guo, Z., Shi, L., Xu, M., Yin, H.: MRCC: A practical covert channel over Monero with provable security. IEEE Access **9**, 31816–31825 (2021). https://doi.org/10.1109/ACCESS.2021.3060285
13. Keller, P., Florian, M., Böhme, R.: Collaborative deanonymization. In: Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin,

DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers. p. 39–46. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-662-63958-0_3

14. Koe, Alonso, K.M., Noether, S.: Zero to Monero: Second edition (2020), https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf
15. 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)
16. Liu, L., Liu, L., Li, B., Zhong, Y., Liao, S., Zhang, L.: MSCCS: A Monero-based security-enhanced covert communication system. Computer Networks **205**, 108759 (2022). https://doi.org/10.1016/j.comnet.2021.108759
17. 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
18. Möller, B.: A public-key encryption scheme with pseudo-random ciphertexts. In: Samarati, P., Ryan, P., Gollmann, D., Molva, R. (eds.) Computer Security – ESORICS 2004. pp. 335–351. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
19. Möser, M., Böhme, R., Breuker, D.: Towards risk scoring of Bitcoin transactions. In: Financial Cryptography and Data Security. pp. 16–32. Springer (2014)
20. 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
21. 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)
22. Persiano, G., Phan, D.H., Yung, M.: Public-key anamorphism in (CCA-secure) public-key encryption and beyond. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024. pp. 422–455. Springer Nature Switzerland, Cham (2024)
23. 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
24. Simmons, G.J.: The prisoners' problem and the subliminal channel. In: Advances in Cryptology: Proceedings of CRYPTO '83. pp. 51–67. Plenum (1983)
25. 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: 29 September 2025
26. Tiemann, T., Berndt, S., Eisenbarth, T., Liskiewicz, M.: "Act natural!": Exchanging private messages on public blockchains. In: 8th IEEE European Symposium on Security and Privacy, EuroS&P 2023, Delft, Netherlands, July 3-7, 2023. pp. 292–308. IEEE (2023). https://doi.org/10.1109/EUROSP57164.2023.00026
27. Zhang, T., Li, B., Zhu, Y., Han, T., Wu, Q.: Covert channels in blockchain and blockchain based covert communication: Overview, state-of-the-art, and future directions. Computer Communications **205**, 136–146 (2023). https://doi.org/10.1016/j.comcom.2023.04.001

A Ring Signatures in Monero

Monero transactions hide input and output amounts a_j using Pedersen commitments $C_j = x_j \cdot G + a_j \cdot H \in \mathbb{G}$ and rely on their being homomorphic to prove balance, i.e., that net input equals net output.[26] When spending some inputs, their commitments' trapdoors x_j are re-randomized, i.e., new commitments C'_j (referred to as *pseudo-output commitments* in [14]) with trapdoors x'_j are constructed, where each C'_j commits to the same amount as the corresponding input commitment C_j, and the output commitments' trapdoors y_t are selected so that the difference $\sum_j x'_j - \sum_t y_t$ vanishes modulo the group order ℓ. If the amounts balance (at least modulo ℓ), then the sum of the pseudo-output commitments must equal the sum of the output commitments, which can be verified by anyone.

What remains to be shown is the consistency of pseudo-output and input commitments. This is proved by showing the knowledge of $z_j = x_j - x'_j$ such that $C_j - C'_j = z_j \cdot G$. This, in turn, is done together with the proof of knowledge of the output key k^o_j (recall Section 2.2) by signing the transaction in question using a Concise Linkable Spontaneous Anonymous Group (CLSAG) signature scheme [11]. CLSAG signatures grant the signer anonymity by using a ring of n public keys with $n-1$ *decoys*, but, importantly, also enable linking of any two signatures produced with the same key $K^o = k^o \cdot G$ by involving a *key image* $\tilde{K} = k^o \cdot \mathcal{H}_p(K^o)$ in the signature. This prevents double-spending, i.e., sending the same asset again in another transaction.

Each transaction input therefore comprises a set of n one-time addresses $\{K^o_1, \ldots, K^o_\pi, \ldots, K^o_n\}$ and n amount commitments $\{C_1, \ldots, C_\pi, \ldots, C_n\}$ with only the πth component in each corresponding to the actual input being spent. The user authorizes the transaction by revealing a pseudo-output commitment C' and signing with k^o_π and z_π where $C_\pi - C' = z_\pi \cdot G$. Write $Z_i = C_i - C'$. To produce a CLSAG signature, the signer chooses $n-1$ random nonces $r_i \in \mathbb{Z}_\ell$ for each $i \in \{1, \ldots, n\}$ excluding $i = \pi$, as well as a single random $\alpha \in \mathbb{Z}_\ell$, and computes *aggregate public keys*

$$W_i = \mathcal{H}_n(1\|\tilde{K}\|\tilde{Z}) \cdot K^o_i + \mathcal{H}_n(2\|\tilde{K}\|\tilde{Z}) \cdot Z_i$$

for each i. A single *aggregate key image* is computed as

$$\tilde{W} = \mathcal{H}_n(1\|\tilde{K}\|\tilde{Z}) \cdot \tilde{K} + \mathcal{H}_n(2\|\tilde{K}\|\tilde{Z}) \cdot \tilde{Z}\,,$$

where $\tilde{K} = k^o_\pi \cdot \mathcal{H}_p(K^o_\pi)$ and $\tilde{Z} = z_\pi \cdot \mathcal{H}_p(K^o_\pi)$. Observe that the signer knows w_π such that $W_\pi = w_\pi \cdot G$ and $\tilde{W} = w_\pi \cdot \mathcal{H}_p(K^o_\pi)$. They proceed by computing

$$c_{(\pi \bmod n)+1} = \mathcal{H}_n\big(M\|(\alpha \cdot G)\|(\alpha \cdot \mathcal{H}_p(K^o_\pi))\big)\,,$$

where M is a hash of the transaction data. The signer then traverses the ring of input transactions, starting with $i = (\pi \bmod n) + 1$, and computes

$$c_{(i \bmod n)+1} = \mathcal{H}_n\big(M\|(r_i \cdot G + c_i \cdot W_i)\|(r_i \cdot \mathcal{H}_p(K^o_i) + c_i \cdot \tilde{W})\big)\,.$$

[26] For clarity of exposition, we neglect the miner's fee.

Finally, they set $r_\pi = \alpha - c_\pi w_\pi \pmod{\ell}$. The signature is

$$\sigma = (c_1, r_1, \ldots, r_n),$$

and the key images $\tilde{K}$ and $\tilde{Z}$ are published alongside it. Importantly for our purposes, out of the n nonces r_i, only one is derived deterministically, namely, r_π. The remaining ones are drawn uniformly at random from $\mathbb{Z}_\ell$. This is the basis for the anamorphic channel in Section 3.

B Anamorphic Cryptography

Anamorphic cryptography [15, 21, 22] is a field concerned with novel uses (or abuses) of established protocols in the presence of a powerful *dictator* who, by rule of law, breaks implicit assumptions underlying modern cryptography such as the secrecy of the private key. Specifically, the dictator may compel participants in the protocol to (at some point) disclose the corresponding secret keys. Anamorphic cryptography studies how the parties may nevertheless achieve their goals (e.g., communicate privately) in this setting without raising the suspicion of the dictator. In particular, the parties must use standard (i.e., legalized and endorsed by the dictator) cryptographic primitives. A protocol Π is said to be anamorphic if it may work in two modes: regular and anamorphic. In the anamorphic mode, the parties use extra *double keys* (the existence of which they should be able to plausibly deny) to achieve some extra goal (e.g., covert communication) under the guise of running Π. Indistinguishability of the two modes of Π, even when the involved parties are compelled to reveal the regular keys, affords the parties deniability.

Relevant to this work is the formal notion of an *anamorphic signature scheme*. The following definition is an adaptation of the one originally due to Kutyłowski et al. [15] to the formalism of [22] which can accommodate the anamorphic channel's being secured with asymmetric cryptography.

Definition 1 (Anamorphic Triplet). *A triplet* $T = (\mathsf{aKeyGen}, \mathsf{aSign}, \mathsf{aDec})$ *of efficient algorithms is an* anamorphic triplet *if*

1. *the* anamorphic key generation *algorithm* $\mathsf{aKeyGen}$*, on input the security parameter* 1^λ*, outputs a pair* $(\mathsf{asvk}, \mathsf{assk})$ *of* anamorphic *keys, the* sender double key sdk*, and the* receiver double key rdk*;*
2. *the* anamorphic signing *algorithm* aSign *takes as input a* regular message msg*, an* anamorphic message amsg*, an anamorphic signing key* assk*, and a sender double key* sdk *and outputs an* anamorphic signature asig*;*
3. *the* anamorphic decryption *algorithm* aDec *takes as input an anamorphic signature* asig *and a receiver double key* rdk *and outputs an anamorphic message* amsg*;*

and there exists a function ε, negligible in the security parameter λ, such that, for every pair of messages $(\mathsf{msg}, \mathsf{amsg})$*,*

$$\Pr\left[\mathsf{aDec}(\mathsf{asig}, \mathsf{rdk}) = \mathsf{amsg} \;\middle|\; \begin{array}{l} \big((\mathsf{asvk}, \mathsf{assk}), (\mathsf{sdk}, \mathsf{rdk})\big) \leftarrow\!\!\$\; \mathsf{aKeyGen}(1^\lambda) \\ \mathsf{asig} \leftarrow\!\!\$\; \mathsf{aSign}(\mathsf{msg}, \mathsf{amsg}, \mathsf{assk}, \mathsf{sdk}) \end{array}\right] \geq 1 - \varepsilon(\lambda)\,,$$

where the probability is taken over the random coin tosses of $\mathsf{aKeyGen}$ *and* aSign*.*

Definition 2 (Anamorphic Signature Scheme). *An unforgeable signature scheme* $S = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* anamorphic *if there exists an anamorphic triplet* $T = (\mathsf{aKeyGen}, \mathsf{aSign}, \mathsf{aDec})$ *such that, for every efficient dictator* $\mathcal{D}$*, there exists a function* ε*, negligible in the security paramter* λ*, such that*

$$|\Pr[\mathsf{RealG}_{S,\mathcal{D}}(\lambda) = 1] - \Pr[\mathsf{AnamorphicG}_{T,\mathcal{D}}(\lambda) = 1] \leq \varepsilon(\lambda)\,,$$

where RealG *is the following game (in which the dictator interacts with the honest signing oracle):*

```
$\mathsf{RealG}_{S,\mathcal{D}}(\lambda)$
$(\mathsf{svk}, \mathsf{ssk}) \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda)$
return $\mathcal{D}^{\mathcal{O}_{\mathsf{real}}(\cdot,\cdot,\mathsf{ssk})}(\mathsf{svk}, \mathsf{ssk})$, where
    $\mathcal{O}_{\mathsf{real}}(\mathsf{msg}, \mathsf{amsg}, \mathsf{ssk}) = \mathsf{Sign}(\mathsf{msg}, \mathsf{ssk})$ ,
```

and $\mathsf{AnamorphicG}$ *is the following game (in which the dictator interacts with the anamorphic oracle):*

```
$\mathsf{AnamorphicG}_{T,\mathcal{D}}(\lambda)$
$\big((\mathsf{asvk}, \mathsf{assk}), (\mathsf{sdk}, \mathsf{rdk})\big) \leftarrow\!\!\$\ \mathsf{aKeyGen}(1^\lambda)$
return $\mathcal{D}^{\mathcal{O}_{\mathsf{anamorphic}}(\cdot,\cdot,\mathsf{assk},\mathsf{sdk})}(\mathsf{asvk}, \mathsf{assk})$, where
    $\mathcal{O}_{\mathsf{anamorphic}}(\mathsf{msg}, \mathsf{amsg}, \mathsf{assk}, \mathsf{sdk}) = \mathsf{aSign}(\mathsf{msg}, \mathsf{amsg}, \mathsf{assk}, \mathsf{sdk})$ .
```

Note that, in both RealG and $\mathsf{AnamorphicG}$, the dictator $\mathcal{D}$ is given the (secret) signing key: ssk and assk, respectively. Definition 2 captures the property of the signature scheme S, that it is possible to produce signatures that look legitimate (i.e., are indistinguishable from ones produced honestly) but carry covert messages amsg addressed to the holder of the receiver double key rdk. For the purposes of this work, it is desirable to also have similar indistinguishability guarantees in the case the dictator has access to the sender double key sdk. A fully-fledged definition is beyond the scope of this work, but the proof of Theorem 1 reveals that the construction of Section 3 is likewise undetectable for a dictator with sdk.