Jochen Schäfer*, Christian Müller, and Frederik Armknecht

If You Like Me, Please Don't "Like" Me: Inferring Vendor Bitcoin Addresses From Positive Reviews

Abstract: Bitcoin and similar cryptocurrencies are becoming increasingly popular as a payment method in both legitimate and illegitimate online markets. Such markets usually deploy a review system that allows users to rate their purchases and help others to determine reliable vendors. Consequently, vendors are interested into accumulating as many positive reviews (likes) as possible and to make these public. However, we present an attack that exploits these publicly available information to identify cryptocurrency addresses potentially belonging to vendors. In its basic variant, it focuses on vendors that reuse their addresses. We also show an extended variant that copes with the case that addresses are used only once. We demonstrate the applicability of the attack by modeling Bitcoin transactions based on vendor reviews of two separate darknet markets and retrieve matching transactions from the blockchain. By doing so, we can identify Bitcoin addresses likely belonging to darknet market vendors.

Keywords: Bitcoin, Markets, Reviews, Identification

DOI 10.2478/popets-2022-0022
Received 2021-05-31; revised 2021-09-15; accepted 2021-09-16.

1 Introduction

1.1 Motivation and Contribution

Over the last years, Bitcoin and similar cryptocurrencies have experienced a sharp increase in popularity. While the privacy implications of Bitcoin in general have already been investigated extensively, the scenario of Bitcoin as means of payment in e-commerce applications is still largely unexplored.

We argue that Bitcoin-based payment systems can cause breaches of privacy in online markets if an attacker gains knowledge about the inner workings of the payment system and the payments that have been processed. To this end, we introduce an attack that allows to identify Bitcoin addresses that are likely to belong to vendors on online markets. As opposed to previous work, our attack is fully automated while relying on *public* information only. More concretely, our contributions are as follows.

Modeling We explain how publicly available information on the time and volume of processed orders obtained from positive vendor reviews and the overall regulations of the market can be modeled. This allows to search the Bitcoin blockchain for candidate addresses and transactions.

Attack Using the aforementioned model, we show how potential Bitcoin addresses of vendors on online markets can be identified. To this end, we first focus on the case that these addresses are used more than once (address reuse). Afterwards, we extend this approach to the case that a vendor uses each payout address only once.

Experimental Validation We demonstrate the practical applicability of these attacks on the two darknet markets Cryptonia Market and Cannazon. By doing so, we are able to identify Bitcoin addresses likely belonging to 308 Cryptonia Market- and 45 Cannazon-vendors, along with the addresses of customers and the markets themselves. We also find multiple instances, in which vendors used the same Bitcoin wallet to receive payments from both markets.

1.2 Structure

In Section 2, we give an overview of related work and explain how our contributions differ. Section 3 briefly explains the technical preliminaries on Bitcoin. In Section 4, we provide our model of online markets (including the structure of ordering processes) and give the considered attacker model. Section 5 contains the de-

***Corresponding Author: Jochen Schäfer:** University of Mannheim, E-mail: jochen.schaefer@uni-mannheim.de
Christian Müller: University of Mannheim, E-mail: christian.mueller@uni-mannheim.de
Frederik Armknecht: University of Mannheim, E-mail: armknecht@uni-mannheim.de

scription of the new attacks, one targeting address reuse (Section 5.2), and one targeting aggregation transactions (Section 5.3). In Section 6, we summarize our experiments that confirm the applicability of our attacks and discuss the findings made. Section 7 concludes the paper.

1.3 Ethical Considerations

Our study relies on human-generated data and has potential implications for the targets of our attack. It should thus be subjected to close scrutiny in ethical evaluation. To do so, we follow the guidelines laid out in the *Menlo Report* [6], one of the most widely used ethics frameworks in IT security research.

Respect for Persons: The primary goal of our attack is to infer Bitcoin addresses of vendors. Depending on the investigated payment system, knowledge of a vendor address also incurs knowledge of addresses belonging to customers and markets. Therefore, it would have been desirable to obtain informed consent from all these affected individuals. In our case however, this was not feasible: The reviews, which form the backbone of our attack, do not reveal their author, giving us no information on who we should obtain consent from. Furthermore, we do not know how many customers there are, who they are or how we could contact them. A similar argument can also be made for the case of vendors and market operators. Conducting research without informed consent can be justified if it involves no more than minimal risk to the subjects and the lack of consent has no adverse effect on the subjects' rights and welfare [6]. We argue that the additional risk our research creates for the subjects is indeed minimal: On direct transfer markets, adversaries can trivially obtain vendor addresses, simply by making test purchases and tracing their payment. The FAQ- and help-sections on both investigated markets explicitly mention this issue and advise users to take the necessary precautions: Referring to Bitcoin, Cannazon states that *"As all information is available to the public, there is a need to obfuscate the journey of your transactions by tumbling your coins."* [2]. Similarly, Cryptonia Market mentions that *"[...] vendors are more vulnerable to the type of attack where an opponent, such as LE agencies, will pose as buyers and attempt to trace a payment to the exchange. This should not be a problem since vendors should ALWAYS anonymize their BTC payments."* [5]. Given these statements, users should be well aware that their transactions and addresses are not secret. As for the reviews, we argue that users deciding to leave feedback do so with the explicit intent to inform others about their experience, meaning that they don't expect secrecy either. The additional risk our attack imposes on the subjects is therefore limited to the fact that addresses can now be obtained passively and for multiple vendors simultaneously.

Beneficence: As mentioned before, the additional risk our work creates for the affected parties is minimal. At the same time, publishing our results enables service providers to design more privacy-aware markets and payment systems. Due to the immutability of the Bitcoin blockchain it is impossible to retroactively mitigate vulnerabilities, which makes it even more important to avoid known weaknesses in the first place. We believe that publishing our work is the best way to ensure that a large number of service operators are made aware of the problem and will thus take the necessary actions, ultimately improving user security. Nevertheless, we will report aggregate statistics and high-level findings only to minimize the risk for the individuals included in our analysis. While we acknowledge that other researchers might want to replicate our work, protecting the interests of our subjects remains our primary concern. Prior to release, we therefore pseudonymize our dataset and strip it of all information that is not strictly required for the replication of our findings.

Justice: Due to the self-selection of the subjects, our analysis focuses on a rather small population, namely the users and operators of darknet markets. However, this population is also the primary beneficiary of our research in the sense that implementing countermeasures against our attack directly improves their privacy. We therefore believe that risk and benefit for the target population are well balanced. If cryptocurrency payments ever see widespread adoption, the benefit of our work could possibly extend to larger parts of the general population. Since this would not cause additional risk to the original population, such a development would not negatively affect the assessment of justice.

Respect for Law and Public Interest: Our research was carried out in compliance with German law. The entire methodology, including data acquisition and analysis, is outlined in the paper and a dataset enabling the replication of our results will be released.

2 Related Work

In the following we provide an overview of existing literature and elaborate how our study differs from previous work.

In a study similar to ours, Goldfeder et al. [9] demonstrate that addresses of users making a Bitcoin payment on an online shop can be tied to personally identifiable information by an external observer based on the order volume and the timestamp of a purchase. This even holds when uncertainty regarding the exact time and transaction volume is introduced. They also show that an attacker observing multiple purchases could even render privacy-preserving techniques like CoinJoin useless. In their threat model, the adversary is supposed to be a third party, e. g., a web tracking service or a service provider, that gets hold of order details via data leaks in the online shop implementation. The privacy implications of their findings are more severe than ours, as they acquire personally identifiable information, whereas our attack only matches vendor profiles to Bitcoin addresses. However, their attacker model is much stronger than ours, as the attacker relies on either cooperating with the online shop or a faulty implementation of the shop system. Also, the attack cannot be carried out ex-post, meaning that the attacker can only learn the Bitcoin addresses if she actively observed the purchases of a user.

Chen et al. [3] investigate the payment systems of darknet markets. The authors perform a descriptive analysis of the payment process for six large cryptomarkets and study the privacy implications of the way the markets handle payments. After making a purchase themselves, they trace their payment to an escrow address of Wall Street Market, from where it is sent to a mixing transaction. They are able to identify the address of the vendor they ordered from by comparing the addresses on the receiving side of the mixing transaction to the time and value of their own purchase and timestamps obtained from reviews. Their approach differs from ours in the sense that it requires an active attacker purchasing a product and sending bitcoin to the vendor.

Jawaheri et al. [10] aim at identifying users of Tor hidden services through their Bitcoin payments. In order to do so, the authors perform large-scale scrapes of Twitter and the `BitcoinTalk.org` discussion forum, from which they extract Bitcoin addresses linked to the respective user profiles on the platform. While their approach proves ties between identified users of online communities and the darknet market Silk Road, it relies on Bitcoin users *deliberately* publishing their personal Bitcoin addresses.

Sabry et al. [16] attempt to identify users of `LocalBitcoins.com`. They obtain publicly available information on completed trades, as well as active and past offers on the website. Their approach is similar to ours, as it exploits the characteristics of a platform and relies on publicly available information only. However, the authors assume that an attacker knows what addresses belong to `LocalBitcoins.com`. While this is a realistic assumption in the case of that specific website, it might not be the case for Bitcoin-based payment systems in general. Also, as both, the linking of Bitcoin transactions to trades on the platform, and the linking of trades to advertisements, is subject to uncertainty, the resulting anonymity set sizes are substantially larger than ours.

Through the application of clustering heuristics, Meiklejohn et al. [15] are able to find multiple larger clusters of Bitcoin addresses in the transaction graph, which they suggest are being controlled by the same entity. They obtain known Bitcoin addresses by either interacting directly with services or by scraping Bitcoin-related websites for relevant information. In doing so, the authors discover flows of illegally obtained bitcoin to cryptocurrency exchanges. Contrary to our work, this study requires either active interaction with the parties under investigation or a deliberate release of the Bitcoin address.

In 2018, the *Center on Sanctions and Illicit Finance* published a memo investigating illicit flows of cryptocurrency into digital currency services [7]. The authors are able to identify regular transactions between illicit entities like darknet markets and, among others, cryptocurrency exchanges, Bitcoin ATMs, and gambling sites. As they used information from a commercial data provider, it remains unclear how exactly the relevant addresses were identified.

To the best of our knowledge, our work differs from previous research in the sense that it produces *one-to-one mappings* between Bitcoin addresses and vendor profiles on a market, while being *fully automated*, using *public information* only and requiring *no interaction* between the attacker and the market or vendors.

3 Bitcoin

We assume that readers are familiar with the basic concepts of Bitcoin and only shortly explain terminology and concepts that are used throughout this paper. As a convention, we refer to the protocol as *Bitcoin* and to the unit of currency as *bitcoin (BTC)*.

3.1 Transactions and Addresses

A *transaction* represents the flow of bitcoin within the Bitcoin network and comprises two lists, transaction inputs and outputs, as well as a unique transaction-ID and further information. Transaction outputs specify a certain amount of bitcoin alongside a set of conditions that must be met if a user wishes to spend the associated coins. Note that each transaction may incur a fee paid for by the sender, which can be calculated as the difference of the input and output sums.

Usually, the spending conditions defined in a transaction output can be represented in the form of a *Bitcoin address*. Addresses are often treated as a kind of user account identifier, as a rational user would never share the data (e. g., private keys) required to spend from it. For the same reason, the multi-input heuristic [15] states, that all addresses jointly occurring on the input-side of a transaction are likely controlled by the same entity.

Address reuse is generally undesirable as it allows for profiling user behavior by outside parties [12, 20]. As there is no limit on the number of addresses per user, they may generate and use as many addresses of any available type as they desire. The overall set of addresses owned by a user is called a user's Bitcoin *wallet*. However, managing many *independent* private keys for a user's wallet might be cumbersome. With the Bitcoin Improvement Proposal 32 (BIP32) so-called Hierarchical Deterministic Wallets were introduced. A single seed can be used to create child keys in a tree-like, deterministic fashion, whereas each child key can be used as a seed for further child keys of its own. Using the key derivation process described in BIP32, anyone with access to a user's extended public key can derive fresh public keys and, thus, addresses that are controlled by the user without any need for interaction between the two [19].

In this paper, we use the term *spending/sending to* an address to describe the presence of a transaction output locking funds to the conditions specified by the address. Correspondingly, *spending/withdrawing from* an address describes cases in which a transaction uses unspent transaction outputs locked to the address-defined conditions as input.

Spending conditions can implement higher-level logic. For instance, users can create m-of-n multisignature (multisig) addresses, meaning that any transaction spending from such an address must be signed by at least m out of n authorized users [1].

3.2 The Bitcoin Graph

In accordance with prior research of Jourdan et al. [11] and McGinn et al. [14], we interpret the Bitcoin ledger as a large, complex graph. Addresses, transactions, and blocks constitute different types of nodes which are connected via directed edges. Figure 1 gives a brief visual overview of how the Bitcoin graph can be modeled in a database schema. The attributes and associated data types are listed in the attached infoboxes, key values are underlined. Addresses, transactions, and blocks all constitute nodes, we only show them in different shapes for clarity. Initially, some interesting values like the sum of incoming and outgoing values of a transaction, the transaction date as well as the in- and out-degrees are only included implicitly in the database. To reduce the number of potentially expensive disk operations and calculations, these values are stored as explicit attributes of the respective nodes. For the description of our attack, only transactions and addresses are relevant. Therefore, we can resort to a more simplistic blockchain model, which mostly ignores the block nodes. Figure 2 shows a hypothetical example of how the currency flow in the

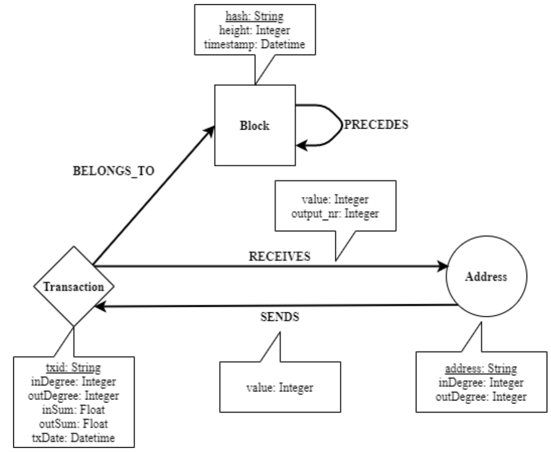


Fig. 1. Database schema (based on Sommer [17]).

Bitcoin graph looks like in our database schema. The nodes and edges shown in red constitute an example of a subgraph.

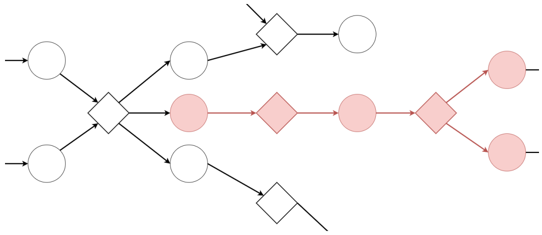


Fig. 2. A hypothetical excerpt of the Bitcoin transaction graph, subgraph shown in red.

4 Model

4.1 Online Markets

We define a market $\mathcal{M}$ as a venue on which a vendor $\mathcal{V}$ sells goods and services to a customer $\mathcal{C}$, where $\mathcal{C}$ and $\mathcal{V}$ also represent a customer's and vendor's Bitcoin wallet consisting of the addresses $\{a^1_{\mathcal{C}} \dots a^c_{\mathcal{C}}\}$ and $\{a^1_{\mathcal{V}} \dots a^v_{\mathcal{V}}\}$.

If a customer $\mathcal{C}$ wants to buy a product from some vendor $\mathcal{V}$, the corresponding order is described as the tuple $(\mathcal{C}, \mathcal{V}, \tau, \alpha) = \vec{o}$, where τ is a timestamp, and α is the total order volume. To pay for the purchase, $\mathcal{C}$ at some point issues a Bitcoin transaction, that we model as $\vec{t} = (A_{\text{in}}, A_{\text{out}}, \tau_{\text{t}}, \alpha_{\text{t}})$, where $A_{\text{in}} = \{a^1_{\text{in}}, \dots, a^i_{\text{in}}\}$ and $A_{\text{out}} = \{a^1_{\text{out}}, \dots, a^o_{\text{out}}\}$ are the sets of input and output addresses, τ_{t} is the timestamp of the transaction, and α_{t} is the transaction volume as defined by the sum of input values. A_{in} encompasses at least one address a_{in} that spends to $\vec{t}$, i. e., inputs a certain amount of bitcoin into $\vec{t}$. Correspondingly, A_{out} includes at least one address receiving bitcoins. The same address can be included in both sets, for example if change is returned to the sender.

4.1.1 Escrow

Unlike in brick-and-mortar stores, where the exchange of money and goods happens simultaneously during checkout, purchases on online stores inevitably cause a circular dependency of trust between $\mathcal{C}$ and $\mathcal{V}$ [8]: $\mathcal{V}$ could ship the ordered product to $\mathcal{C}$ prior to receiving payment, or $\mathcal{C}$ could pay $\mathcal{V}$ before receiving the product. Either way, the party moving first has to trust that the counterparty behaves honestly.

This problem can be remedied if $\mathcal{M}$ is able to act as a trusted third party [8]. In order to do so, $\mathcal{M}$ provides an escrow payment system that guarantees fairness to both, $\mathcal{C}$ and $\mathcal{V}$. That is, $\mathcal{C}$ makes the payment first to some deposit address a_{dep}. $\mathcal{M}$ then holds the funds in escrow until the order is completed. In case of problems, $\mathcal{C}$ can file a complaint with $\mathcal{M}$, who then mediates a solution. If no problems arise, the payment is released, i. e., it is eventually transferred to some vendor address.

Depending on the exact implementation $\mathcal{M}$ may use a dedicated escrow address a_{esc}, although the funds can also be held on the deposit address, in which case $a_{\text{dep}} = a_{\text{esc}}$.

Typically, $\mathcal{M}$ receives compensation for its services, usually in form of a fee φ. Such a fee could be calculated in many different ways, but a straightforward approach would be to charge fees based on the order volume α. The fees $\mathcal{V}$ has to pay might differ by product category, vendor type or other factors. If $\mathrm{p}_{\min}$ and $\mathrm{p}_{\max}$ are the minimum and maximum percentages $\mathcal{M}$ charges as a commission based on the order volume, φ should fall in the range $\varphi_{\min} = \mathrm{p}_{\min} \cdot \alpha \leq \varphi \leq \mathrm{p}_{\max} \cdot \alpha = \varphi_{\max}$, where $\varphi_{\min}$ and $\varphi_{\max}$ represent the minimum and maximum fee $\mathcal{M}$ charges, respectively.

4.1.2 Payment Systems

The observations of Chen et al. [3] indicate, that the payment systems of cryptomarkets differ in two aspects: Firstly, whether they require customers to make deposits upfront, and secondly, whether they cause a direct flow of currency between customers and vendors. We use this observation to introduce the following new terminology.

In *wallet-based* markets, the deposit address a_{dep} is associated with the customer $\mathcal{C}$, meaning that each customer has at least one unique market-supplied Bitcoin address. Any funds sent to a_{dep} are credited to $\mathcal{C}$'s account on $\mathcal{M}$ and can be used to make purchases. When placing an order $\vec{o}$, funds amounting to the total order volume α are debited from $\mathcal{C}$'s account, held in escrow, and credited to $\mathcal{V}$'s account upon successful completion. In *walletless* markets, a_{dep} is associated with the order $\vec{o}$ and hence used for one purchase only. Customers do not have store credit, but have to send the required amount of bitcoin to a deposit address every time they place an order.

In *direct transfer* markets, funds on a_{dep} are transferred directly to the vendor, causing a direct connection between $\mathcal{C}$ and $\mathcal{V}$. *Centralized transfer* markets on the other hand handle payments via a market wallet. Although vendors receive the appropriate amount of bitcoin, the coins can be different and there is no direct flow of currency between customer and vendor. This also implies that the deposits of $\mathcal{C}$ and the withdrawals of $\mathcal{V}$ do not necessarily correspond to a specific order: To save Bitcoin fees, $\mathcal{C}$ might deposit funds for multiple purchases at once, just as $\mathcal{V}$ could prefer to withdraw the revenue of multiple sales in one transaction. Also, there are many different ways in which a market could implement a centralized transfer payment system, which makes it hard to develop a general description of such systems. Because of that, this study focuses on direct transfer payment systems and their specific characteristics.

For instance, the processing of an order on a *walletless direct transfer market* follows a series of six consecutive steps:

1. Customer $\mathcal{C}$ creates order $\vec{o} = (\mathcal{C}, \mathcal{V}, \tau, \alpha)$ on market $\mathcal{M}$.
2. $\mathcal{C}$ receives payment instructions from $\mathcal{M}$.
3. $\mathcal{C}$ generates a *deposit transaction* $\vec{t}_{\text{dep}}$ from $a_{\mathcal{C}} \in \mathcal{C}$ to a_{dep} over amount α_{dep} ($\alpha_{\text{dep}} \geq \alpha +$Bitcoin fee). If a dedicated escrow address is used, i. e., $a_{\text{dep}} \neq a_{\text{esc}}$, $\mathcal{M}$ issues a *transfer transaction* $\vec{t}_{\text{tra}}$ to move the funds to a_{esc}.
4. $\mathcal{V}$ fulfills the order, e. g., by shipping the ordered goods.
5. $\mathcal{C}$ receives the purchase and finalizes the order.
6. $\mathcal{M}$ generates a *payout transaction* $\vec{t}_{\text{pay}}$ from a_{esc} to $\mathcal{V}$.

4.1.3 Structural Properties of Transactions

Assuming that all involved parties act rationally, the payment process incurs certain structural properties of the corresponding transactions. The deposit transaction $\vec{t}_{\text{dep}}$ should have exactly one output spending to a_{dep} but might have other outputs spending to other addresses. The transaction output spending to a_{dep} is expected to have a value greater than or equal to α. If a market uses dedicated escrow addresses, we would expect a transfer transaction $\vec{t}_{\text{tra}}$, having exactly one input spending from a_{dep}, one output spending to a_{esc} and potentially a second output transferring excess deposits back to $\mathcal{C}$. In case a_{esc} is a 2-3 multisig address, an additional output is to be expected, sending the fee φ to some address $a_{\mathcal{M}}$ controlled by $\mathcal{M}$. This is because a 2-3 multisig escrow address would allow $\mathcal{C}$ and $\mathcal{V}$ to cooperatively issue a payout transaction that deprives the market of its fees. A standard payout transaction $\vec{t}_{\text{pay}}$ is expected to have at least one input spending from a_{esc} and at least one, but at most as many outputs as there are parties involved in the trade. One of these outputs is transferring the vendor's revenue to a vendor address $a_{\mathcal{V}} \in \mathcal{V}$. A second output sending φ to commission address $a_{\mathcal{M}}$ may occur if $a_{\text{dep}} = a_{\text{esc}}$, since in this case $\vec{t}_{\text{pay}}$ is the only transaction controlled entirely by the market. As with $\vec{t}_{\text{tra}}$, an optional third output could occur if excess deposits are transferred back to $\mathcal{C}$. While $\vec{t}_{\text{pay}}$ could theoretically have even more outputs, there is no obvious reason why more recipients should exist. To save transaction fees, payouts from multiple escrow addresses of a vendor can be batched in a single $\vec{t}_{\text{pay}}$. As a_{dep} and a_{esc} are order-specific and thus used only once, they both should have an in- and outdegree of 1. Possible exceptions with higher indegrees may occur if $\mathcal{C}$ splits the deposit across multiple transactions, either by mistake or to avoid address clustering. Since such a split would result in increased total transaction fees, we assume that the number of split deposits is negligible.

4.1.4 Modeling the Transaction Patterns

Using these information, it is possible to model the transaction patterns of a payment system. For instance, a walletless direct transfer payment system that does not deploy a dedicated escrow address can be described by the simple directed graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ that represents the system's currency flow on the blockchain. The vertices of this graph are $\mathrm{V} = \{a_{\mathcal{C}}, a_{\text{dep}}, a_{\mathcal{V}}, a_{\mathcal{M}}, \vec{t}_{\text{dep}}, \vec{t}_{\text{pay}}\}$. The set of edges is $\mathrm{E} = \{\left(a_{\mathcal{C}}, \vec{t}_{\text{dep}}\right), \left(\vec{t}_{\text{dep}}, a_{\text{dep}}\right), \left(a_{\text{dep}}, \vec{t}_{\text{pay}}\right), \left(\vec{t}_{\text{pay}}, a_{\mathcal{M}}\right), \left(\vec{t}_{\text{pay}}, a_{\mathcal{V}}\right)\}$. The transaction nodes $\vec{t}_{\text{dep}}$ and $\vec{t}_{\text{pay}}$ are also associated with the timestamps τ_{dep} and τ_{pay}, derived from the block they were included in. Edges are associated with a value α_{in} or α_{out}, specifying the amount of bitcoin they transfer. Figure 3 is a visual representation of said graph.

In cases in which a dedicated escrow address is used, the transaction pattern becomes slightly more complex. As shown in Figure 4, the corresponding graph $\mathrm{G}^* = (\mathrm{V}^*, \mathrm{E}^*)$ encompasses two additional nodes, such that $\mathrm{V}^* = \{a_{\mathcal{C}}, a_{\text{dep}}, a_{\text{esc}}, a_{\mathcal{V}}, a_{\mathcal{M}}, \vec{t}_{\text{dep}}, \vec{t}_{\text{tra}}, \vec{t}_{\text{pay}}\}$ and $\mathrm{E}^* = \{\left(a_{\mathcal{C}}, \vec{t}_{\text{dep}}\right), \left(\vec{t}_{\text{dep}}, a_{\text{dep}}\right), \left(a_{\text{dep}}, \vec{t}_{\text{tra}}\right), \left(\vec{t}_{\text{tra}}, a_{\text{esc}}\right), \left(\vec{t}_{\text{tra}}, a_{\mathcal{M}}\right) \left(a_{\text{esc}}, \vec{t}_{\text{pay}}\right), \left(\vec{t}_{\text{pay}}, a_{\mathcal{M}}\right), \left(\vec{t}_{\text{pay}}, a_{\mathcal{V}}\right)\}$. Note

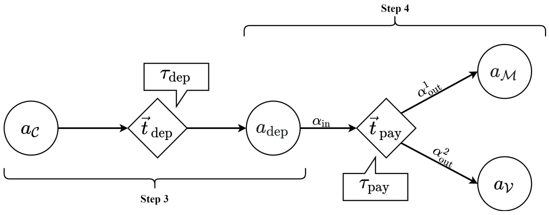


Fig. 3. A payment system subgraph where $a_{\text{dep}} = a_{\text{esc}}$.

that the two edges connecting $\vec{t}_{\text{tra}}$ and $\vec{t}_{\text{pay}}$ to $a_{\mathcal{M}}$ are mutually exclusive.

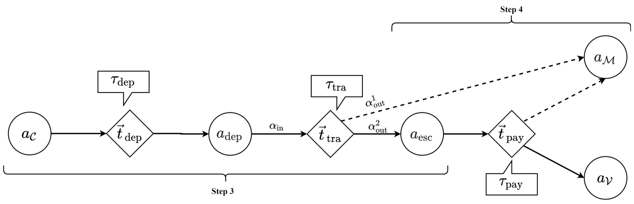


Fig. 4. A payment system subgraph where $a_{\text{dep}} \neq a_{\text{esc}}$.

4.1.5 Matching Graphs

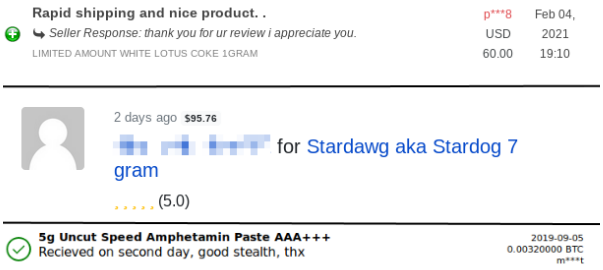
In our attack, we will use this structure of G to identify candidate transactions in the Bitcoin graph. From now on, we refer to any instance of G, i.e., a graph having the structure explained above, where all nodes and edges of G are a subset of the overall Bitcoin transaction graph, as a subgraph *matching the payment system of market* $\mathcal{M}$. For example, the subgraph highlighted in Figure 2 *matches* the simple walletless direct transfer payment system described above. In principle, the graph G is also applicable for wallet-based direct transfer markets. In such a case, the node $\mathcal{C}$ would be replaced with a market wallet address and $\vec{t}_{\text{dep}}$ would be issued by $\mathcal{M}$.

Besides matching the structural properties, it is important for our attack to filter transactions also according to their content. In practice, limited information and systematic uncertainty may prevent us from knowing the precise transaction values. For example, τ_{pay} is derived from the timestamp of the block $\vec{t}_{\text{pay}}$ is included in. However, new blocks are not created uniformly and it is not guaranteed that a transaction is immediately included in the next block mined. This leads to an unknown delay between the finalization of the order and the timestamp of the payout transaction τ_{pay}. Uncertainty may also arise from limited information, for instance if the volume of an order is only available in a fiat currency and the exchange rate is unknown. For these reasons, we extend the meaning of matching graphs to *approximately match* (partial) information about certain transactions. Formally, this means that the absolute difference of certain numerical values does not exceed a specified threshold.

4.1.6 Reviews

Even with an escrow system in place, customers still have to trust vendors that they sell products of appropriate quality. This can be addressed by a review system, in which $\mathcal{C}$ can create reviews for vendors and/or products.

A review is at least associated with the rated vendor or product, but additional credibility can be achieved if a review is tied to an order $\vec{o}$, i. e., only customers who have actually bought a product can leave feedback. Figure 5 shows an example of what reviews look like on three different darknet markets.


Rapid shipping and nice product. . p***8 Feb 04, 2021
↪ Seller Response: thank you for ur review i appreciate you. USD 60.00 19:10
LIMITED AMOUNT WHITE LOTUS COKE 1GRAM

2 days ago $95.76
[redacted] for Stardawg aka Stardog 7 gram
(5.0)

5g Uncut Speed Amphetamin Paste AAA+++ 2019-09-05
Recieved on second day, good stealth, thx 0.00320000 BTC m***t

Fig. 5. Examples of positive reviews on three darknet markets

4.2 Attacker

We assume an attacker whose goal it is to identify the Bitcoin addresses interacting with a given online shop, namely the vendors, the customers, and the market wallet. The possible motivations of such an attacker to target payment privacy are manifold. Legitimate reasons may include law enforcement agencies or governments tackling tax evasion and money laundering or tracking down funds obtained from criminal activities. In such scenarios, the ability to passively identify addresses is particularly useful, as it effectively prevents accusations of entrapment during the criminal investigation.

Depending on the use of the Bitcoin addresses and the granularity of the data, the payment history might also reveal personal information of customers such as financial status, shopping habits, or a list of purchased products, all of which could be of interest for advertisers, credit agencies, insurances and the like. Finally, competitors of the targeted market and the vendors could use the data to infer information on approximate sales volumes and profits.

Regarding attacker capabilities, the attacker is strictly limited to public data. This includes the Bitcoin blockchain as well as any publicly accessible website on the Internet and overlay networks like Tor.

5 Attacks

In this section, we present an attack that aims for identifying the Bitcoin addresses of vendors in online markets, using public information only. In a nutshell, the attack consists of finding and combining Bitcoin addresses in the Bitcoin database that meet *structural* and *content-related* criteria.

Regarding the structure, addresses should occur at the position of the vendor payout address in at least one payment system subgraph matching the market under investigation.

Regarding the content, we aim to map orders to transactions. Therefore, the subgraphs do not only have to match the structural characteristics of the payment system, but also the properties of the order that triggered the payment.

5.1 Overview

It is known from prior research that adversaries can identify a Bitcoin transaction and therefore the spending and receiving addresses if the precise timestamp τ and order volume α of the transaction are known [9]. To this extent, our attack relies on the assumption that an adversary can compensate for imprecise information by inferring the structure of a market's payment system and obtaining estimates of τ and α for multiple transactions. The less precise the estimates are, the more distinct observations are required.

There are several scenarios imaginable of how an attacker could obtain such estimates: Affiliate marketing users, loyalty programs or even bystanders observing purchases on a victim's computer screen could all get access to the approximate value and time of purchases paid for with bitcoin. As these are all relatively strong attackers, we resort to more accessible information and use *positive* vendor reviews as our source of data.

In order to be useful, a review must at least contain some information α_r on the actual order volume α and a timestamp τ_r at which it was posted. Such a review can be represented by the tuple $\vec{r} = (\tau_\mathrm{r}, \alpha_\mathrm{r})$. As of June 2021, 4 out of 15 darknet markets investigated by us featured such reviews and another 2 provided intervals of the values in question, making this a fairly reasonable assumption.

It is also assumed that the attacker can obtain an estimate of the parameter $\delta = |\tau_\mathrm{r} - \tau_\mathrm{pay}|$, which is the time difference between the occurrence of a payout transaction and the publishing of the corresponding review. An example of how such an estimate can be obtained is provided in Section 6.3. The interval $[\tau_\mathrm{r} - \delta, \tau_\mathrm{r} + \delta]$ can then serve as an estimate for τ_pay, the time at which $\vec{t}_\mathrm{pay}$ occurs.

The amount $\vec{t}_\mathrm{pay}$ is transferring, α_pay, can be estimated by the review amount α_r.

Nonetheless, there are situations in which reviews do not provide useful estimates. For instance, negative reviews cannot be used to estimate transaction times, as they might be related to a dispute. Disputes could cause the funds to be held in escrow for an extended period of time or to be refunded to the customer after all. Using reviews as an estimate for transaction times is also not possible if the vendor is allowed to finalize early, i. e., is allowed to demand advance payments. In this case, the payout transaction precedes the review by an unknown time span. Some markets also allow to edit reviews: For instance, an unsatisfied customer could initially leave negative feedback and open a dispute. If the two parties reach an amicable settlement, the customer could revise the review to be positive. An attacker, who is unaware of this process, would now unsuccessfully search for a payout transaction at the time of the review. Finally, reviews may not always indicate which cryptocurrency the customer paid the order in. If a vendor accepts multiple currencies, the transactions corresponding to the reviews could be divided across different blockchains.

To carry out the attack, an adversary would first acquire as many vendor reviews from the market as possible. Then, the attacker defines a payment system graph representing the structure of the market's payment system. Once the payment system graph and estimates for timestamps and transaction volumes are known, the attacker begins to search the Bitcoin transaction graph for subgraphs that match the payment system and the reviews. At this point, the attack differs depending on the presence of address reuse.

5.2 Targeting Address Reuse

In the following, we detail the attack for the case that a vendor uses the same vendor payout address $a_{\mathcal{V}} \in \mathcal{V}$ multiple times.

For this purpose, the attacker iterates over the vendor reviews and searches the Bitcoin graph for subgraphs matching the estimates of α and τ obtained from the review. The attacker proceeds to extract the addresses of $\mathcal{V}$ from the matched subgraphs, which yields the set of potential payout addresses, A_{cand}, for a given review $\vec{r}$ (see line 3 of Algorithm 1). The corresponding method `queryDatabase` takes a review $\vec{r}$, the review-transaction-delay δ and the marketplace fee φ as parameters, retrieves matching subgraphs as defined in Section 4.1.5 and returns the set of addresses occurring at the position of $a_{\mathcal{V}}$. From the second review on, the set of potential payout addresses of the current iteration is intersected with those of the previous iterations (see line 7). If the intersection with the current candidate set would result in the empty set, the iteration is skipped (see line 6). The process stops once a candidate address has been found, i. e., the intersection results in a set of size 1, or if there are no more reviews left to process. Algorithm 1 describes this process in pseudocode.

Algorithm 1: Linkage attack for address reuse

```
   Input: $R$: All positive reviews for a fixed
          vendor $\mathcal{V}$
   Result: A candidate address set containing
           one candidate address.
1  $A_{\text{cand}} := \emptyset$
2  Chronologically select next $\vec{r}^*$ from $R$
3  $A_{\text{cand}} :=$ queryDatabase($\varphi, \delta, \vec{r}^*$)
4  for all $\vec{r} \in R \setminus \{\vec{r}^*\}$ do
5      $A'_{\text{cand}} :=$ queryDatabase($\varphi, \delta, \vec{r}$)
6      if $|A_{\text{cand}} \cap A'_{\text{cand}}| > 0$ then
7          $A_{\text{cand}} := A_{\text{cand}} \cap A'_{\text{cand}}$
8      if $|A_{\text{cand}}| = 1$ then
9          break
10 return $A_{\text{cand}}$
```

The set of candidate addresses returned by Algorithm 1 may contain false positives. For instance, the algorithm could match addresses that are unrelated to the vendor and simply exhibit some matching transaction patterns by chance. Another source of false positives are vendors with overlapping trading activity on the same market. Since there are multiple vendors offering similar products at similar prices, the same address may match two or more vendors at the same time. To reduce the number of false positives as much as possible, candidate addresses undergo an additional validation step, which is executed after Algorithm 1 and works as follows.

Given a candidate address for a vendor, the attacker queries the database for all payment system subgraphs in which the candidate address occurs at the position of $a_{\mathcal{V}}$ (see Figure 7). The attacker then creates the set W by extracting the tuple $w = (\alpha_{\text{pay}}, \tau_{\text{pay}})$ for each of those graphs, where α_{pay} is the amount spent by $\vec{t}_{\text{pay}}$ and τ_{pay} is the timestamp of $\vec{t}_{\text{pay}}$. In case the precision of these values differs from the one of $\vec{r}$, they have to be adjusted to the precision of the lower-information data and/or converted to the same currency. For example, the blockchain data contains UNIX timestamps with a resolution of one second and transaction values with a precision of eight decimal places. If the acquired reviews would only list the date on which they were published and the order volume with a precision of four decimal places, the precision of the blockchain-derived values in w has to be reduced accordingly. This means that, if a matching payment system subgraph exists for a review $\vec{r}$, it should also hold that there exists $w \in W$ such that $w = \vec{r}$. The set of payment system subgraphs for which a matching review exists is thus defined as $M = W \cap R$.

To evaluate how well a candidate address fits a vendor, the similarity between the observed transactions and the reviews of a vendor can be calculated. A straightforward measure would be $J(W, R)$, the Jaccard index between the observed and expected payment system subgraphs for a given vendor-address-pair. In general, the Jaccard index $J(X, Y)$ is a similarity measure for two non-empty finite sets X, Y and is computed by dividing the size of the intersection of the sets by the size of their union. The result can be heuristically interpreted as the probability that an element of at least one of the sets is an element of both [13]. To leverage this interpretation in a human-friendly way, we actually calculate two separate Jaccard similarities, namely $J(M, R)$ and $J(M, W)$, where $M \subseteq R$ and $M \subseteq W$. Thus, $J(M, R)$ is the probability that, given a vendor review, a matching payment system subgraph spending to the investigated address exists. To better underline the meaning of this metric, we refer to it as review coverage (RC):

$$\mathrm{RC} = J(M, R) = \frac{|M \cap R|}{|M \cup R|} = \frac{|M|}{|R|} \tag{1}$$

Similarly, address coverage (AC) is the probability that a matching review exists for a potential payout transaction spending to the investigated address:

$$\mathrm{AC} = J(M, W) = \frac{|M \cap W|}{|M \cup W|} = \frac{|M|}{|W|} \tag{2}$$

Review coverage and address coverage alone do not prevent false positives, as vendors with very few reviews or candidate addresses included in very few payment system subgraphs could trivially achieve high coverage on either of them. Borrowing the idea of the *F-Score*, we therefore combine the two coverages into the combined coverage score (CCS), which is defined as the harmonic mean of review and address coverage:

$$\mathrm{CCS} = \frac{2 \cdot \mathrm{RC} \cdot \mathrm{AC}}{\mathrm{RC} + \mathrm{AC}} \tag{3}$$

An attacker can now choose a CCS threshold based on how much uncertainty is acceptable. Only addresses that are sufficiently similar, i. e., addresses that have a CCS greater than or equal to the chosen threshold are accepted as likely vendor payout addresses. If an address still matches two separate vendors, even after thresholding, the use of an unified criterion like the CCS allows to determine the most likely match by directly comparing the individual vendor-address-fits.

5.3 Targeting Aggregation Transactions

If $\mathcal{V}$ strictly avoids address reuse and changes $a_{\mathcal{V}}$ for every order, the approach mentioned above most likely fails. In such a case, an attacker could only identify $a_{\mathcal{V}}$ if the estimates of order volume and the timestamp were so specific, that only a single address in the entire blockchain would fit these criteria.

However, constantly generating new payout addresses also implies that $\mathcal{V}$ ends up with funds scattered across several individual addresses. If $\mathcal{V}$ chooses to spend an amount of bitcoin exceeding the balance of any single one of these addresses, the corresponding transaction must aggregate as many inputs from different vendor-controlled addresses as are needed to match or surpass the transaction value. This becomes apparent in an aggregation transaction $\vec{t}_{\mathrm{agg}}$, which combines inputs from at least two different addresses. Figure 6 demonstrates how $\vec{t}_{\mathrm{agg}}$ would appear in the Bitcoin graph.

If $\vec{t}_{\mathrm{agg}}$ aggregates inputs from vendor payout addresses, each of these addresses is also part of a payment system subgraph. Figure 6 can therefore be seen as an expansion of Figure 3, where at least some of the input addresses of $\vec{t}_{\mathrm{agg}}$ are vendor payout addresses of a payment system subgraph. An attacker can use this property to identify one-time vendor payout addresses. As shown in Algorithm 2, `queryDatabase` is called for every review of a vendor to search the database for subgraphs that fit the expected pattern, date, and volume (see line 3). The attacker then subsets the receiving addresses A' to those having an in- and outdegree of exactly 1 (see lines 5 and 6). In the next step, the set T is obtained, which consists of all transactions $\vec{t}_{\mathrm{agg}}$ that have at least one receiving address $a' \in A'$ spending to them (see line 7). The set of candidate addresses is then defined as the set of all addresses spending to a transaction $\vec{t}_{\mathrm{agg}} \in T$, where $\vec{t}_{\mathrm{agg}}$ has at least *minInputs* inputs and a proportion of at least *threshold* of those inputs can be matched to reviews (see lines 10 and 11). The higher the two thresholds are, the lower is the risk for false positives. As stated by the multi-input heuristic (cf. Section 3.1), all addresses spending to such an aggregation transaction likely belong to the same vendor.

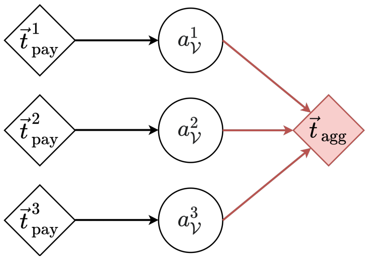

$\vec{t}^{\,1}_{\mathrm{pay}}$
$\vec{t}^{\,2}_{\mathrm{pay}}$
$\vec{t}^{\,3}_{\mathrm{pay}}$
$a^1_{\mathcal{V}}$
$a^2_{\mathcal{V}}$
$a^3_{\mathcal{V}}$
$\vec{t}_{\mathrm{agg}}$

Fig. 6. Vendor addresses spending to aggregation transaction (shown in red).

5.4 Vulnerable Markets

Not all markets are equally vulnerable to our attack. For instance, markets deploying a wallet-based centralized transfer payment system are hard to attack, as customers and vendors may make arbitrary deposits and withdrawals that do not necessarily match any single order. On walletless centralized transfer markets, an attacker could learn deposit addresses and, thus, customer addresses if a customer pays multiple orders from the same wallet. Direct transfer markets, however, are likely vulnerable, regardless of whether they are wallet-based, as a transaction from the customer to the vendor is expected to occur every time an order is finalized. For the purpose of our experiment, we consider all direct transfer markets to be vulnerable, if estimates for α and τ_{pay} can be obtained. Any additional information on the pay-

Algorithm 2: Linkage attack without address reuse

```
Input: $R$: All positive reviews for a fixed
       vendor $\mathcal{V}$, minInputs: Minimum
       number of inputs an aggregation
       transaction needs to have, threshold:
       proportion of inputs needed to have a
       matching review
Result: The set of candidate addresses
        involved in aggregation transactions
$A := \emptyset$
for all $\vec{r} \in R$ do
    $A' :=$ queryDatabase$(\varphi, \delta, \vec{r})$
    for all $a' \in A'$ do
        if indegree$(a') = 1 \wedge$ outdegree$(a') = 1$
         then
            $A := A \cup \{a'\}$
$T = \left\{\vec{t} = (A_{\text{in}}, A_{\text{out}}, \tau_{\text{t}}, \alpha_{\text{t}}) \,|\, \exists a \in A' : a \in A_{\text{in}}\right\}$
$A_{\text{cand}} := \emptyset$
for all $\vec{t} = (A_{\text{in}}, A_{\text{out}}, \tau_{\text{t}}, \alpha_{\text{t}}) \in T$ do
    if $|A_{\text{in}}| \geq$ minInputs $\wedge \frac{|A_{\text{in}} \cap A|}{|A_{\text{in}}|} \geq$ threshold
     then
        $A_{\text{cand}} := A_{\text{cand}} \cup A_{\text{in}}$
return $A_{\text{cand}}$
```

ment system are likely to increase the success probability.

6 Experimental Evaluation

6.1 Overview

In this section, we demonstrate the applicability of our attack by running it against two real-world darknet markets: Cryptonia Market and Cannazon. For this purpose, we retrieved Bitcoin data by running a *full node* and letting the node synchronize up to the current height of the blockchain (the latest included block was at height 662,472 with a block time of 2020-12-22, 09:14 UTC). The binary blockchain data was then parsed and converted into a format that would yield the schema presented in Figure 1 when fed into a Neo4j graph database.

6.2 Environment

The initial experiment in this paper was carried out starting mid-2019 for Cryptonia Market. Being a direct transfer market [4], Cryptonia Market readily reveals information on the properties of the payment process in their FAQ section (see Appendix A). Moreover, the vendor reviews listed on their website also feature a timestamp and the order volume in bitcoin with a precision of four decimal places (truncated, not rounded). It should be noted that Cryptonia Market is no longer accessible, as it ceased operations without further explanation.

For the purpose of validating our methodology, we decided to replicate our experiments at a later point in time and on a different market. As of June 2021, the darknet price search engine *dread* listed a total of 15 active darknet markets. Further evaluation of these markets showed that at least three of them (Cannazon, CannaHome and The Versus Project) made use of direct transfer payment systems. We decided to focus on Cannazon, as this market provides the necessary reviews and details on its payment system (see Appendix A). Furthermore, Cannazon appeared to be the oldest of the three markets and was operational at the same time as Cryptonia Market, which allowed to investigate whether the same payout addresses were used by vendors on both markets.

In accordance with our attacker model (cf. Section 4.2), we systematically scraped all information *publicly* accessible on the Cannazon and Cryptonia Market websites. The webscraper was implemented using the *Selenium* library for Python and great care was taken not to interfere with the regular operation of the markets. The scrape includes all data published on item-listing-, vendor-profile-, and review-pages. Furthermore, screenshots of all accessed pages were taken for documentation purposes, and all product images as well as vendor profile pictures were downloaded. A subset of our data is available for download online[1].

6.2.1 Cryptonia's Payment System

The help- and FAQ-pages of Cryptonia Market provide a rather detailed description of the payment system. By analyzing the FAQ section we can learn that:

- Being a walletless market, Cryptonia Market generates a unique Bitcoin deposit address a_{dep} for every order $\vec{o}$.
- a_{dep} also serves as escrow address a_{esc}.

[1] https://www.wim.uni-mannheim.de/ths/research/data

- Funds from this a_{dep} are transferred to the vendor's payout address $a \in \mathcal{V}$ within minutes after the order has been finalized.
- The payout address has to be configured on the vendor settings page. If the vendor specifies a Bitcoin address, this address receives all incoming payments until a new one is set up.
- Vendors may choose to provide a BIP32 extended public key instead of a payout address. In that case, a new and unused payout address is generated for every order.
- Excess funds on escrow addresses are refunded to the customer if they exceed 0.00005 bitcoin.
- Customers are able to review a vendor immediately after finalizing the order.
- Cryptonia Market's fees are sampled at random from the range between $\mathrm{p}_{\min} = 2\%$ and $\mathrm{p}_{\max} = 4\%$ of the total order volume α.
- Coins are transferred directly from the customer to the vendor. No mixing is performed in between.

The FAQ section is not entirely clear about how transaction fees are handled. As the item prices refer to the amount of bitcoin that has to be deposited on a_{dep}, the transaction fees of $\vec{t}_{\text{dep}}$ should be paid by $\mathcal{C}$. With regards to $\vec{t}_{\text{pay}}$, we interpret the terms so that the market commission φ is calculated based on the total order volume α, regardless of the payout transaction fee. This means that the amount of the output spending to $a_{\mathcal{M}}$ should always be in the range of 2% to 4% of the total order volume, while the output spending to $\mathcal{V}$ corresponds to the remaining 96% to 98% of the total order volume α minus the Bitcoin transaction fee for $\vec{t}_{\text{pay}}$.

Figure 7 shows the pattern we would expect from a Cryptonia Market payout transaction. During matching, the values marked in orange can be estimated from reviews, while the market commission (marked in blue) can be calculated from the review and the information provided on the FAQ page. We use these additional constraints to narrow down the number of matching subgraphs during the attack.

Since there is no incentive for a customer to overpay orders, we assume that the majority of payout transactions has two outputs. Matching overpaid orders would be fairly difficult anyway, as the input value of $\vec{t}_{\text{pay}}$ might no longer match the order volume stated in the review and there is no way of knowing by how much an order was overpaid.

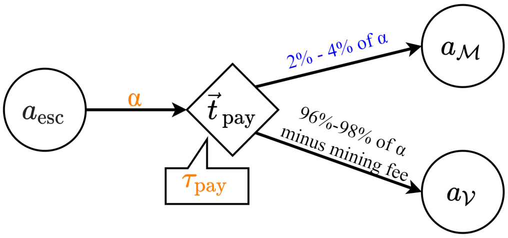


Fig. 7. Expected currency flow on Cryptonia Market. Orange values can be estimated from review, blue values can be calculated using additional information, black values are unknown.

6.2.2 Cannazon's Payment System

Cannazon provides a fairly detailed description of the order and payment process in its help section as well. We identify the following key differences with respect to Cryptonia Market:

- The structure of the payment system differs for escrow- and finalize early-transactions.
- If escrow is used, a transfer transaction $\vec{t}_{\text{tra}}$ sends the funds from a_{dep} to multisig escrow address a_{esc}. Otherwise, funds are transferred directly to $\mathcal{V}$ in $\vec{t}_{\text{pay}}$.
- Market fees are deducted in $\vec{t}_{\text{tra}}$ for escrow orders and in $\vec{t}_{\text{pay}}$ when finalize early is used.
- Because of multisig escrow, vendors have to download, sign and broadcast the transactions spending from a_{esc} manually.

Cannazon actively encourages users to leave feedback. Failure to do so might bar users from placing new orders. Unlike Cryptonia Market, where market fees are sampled at random, Cannazon calculates the commission based on vendor activity and reputation. For every completed order, the vendor is awarded a certain amount of *level points* lp which is calculated as follows (See Figure 18 in the appendix):

$$\mathrm{lp} = 2\alpha + x_{\text{pos}} \cdot \alpha - x_{\text{neg}} \cdot 5\alpha - x_{\text{dis}} \cdot 10\alpha, \qquad (4)$$

where $x_{\text{pos}}, x_{\text{neg}}, x_{\text{dis}} \in \{0, 1\}$ are binary variables indicating whether the customer left a positive or negative review or won a dispute against the vendor. The more level points a vendor accumulates, the lower the market fee becomes. A table showing the individual level point thresholds can be found in Figure 18 in the appendix.

If customers fail to inform the market that they have received their merchandise, orders will auto-finalize three days after the expected delivery date. This imposes an upper bound on time elapsed between τ_{tra} and τ_{pay}.

Figure 8 visualizes the currency flow associated with a Cannazon escrow transaction.

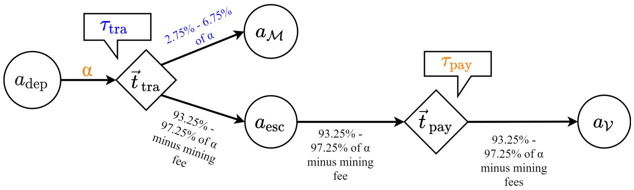


Fig. 8. Expected currency flow on Cannazon. Orange values can be estimated from review, blue values can be calculated using additional information, black values are unknown.

6.3 Experiment

We run our attacks against both markets, using a total of 28,966 reviews from Cryptonia Market and 351,769 from Cannazon. As the reviews on Cryptonia Market specify the currency, we exclude orders that were paid in Monero. The same applies to reviews on Cannazon if they explicitly refer to finalize-early transactions, because τ_{r} cannot be used to estimate τ_{pay} in these cases. All Cryptonia Market reviews, as well as Cannazon reviews with unknown transaction type are treated as escrow transactions. During the attack, we process the reviews in chronological order, starting with the most recent ones. Once a matching address is found, the date of the first incoming transaction of that address is retrieved. The matching process is repeated until there are no reviews left that are older than the last matched address.

Since Cannazon reviews state the transaction volume in Euro, we have to convert the amounts to bitcoin prior to matching. This causes some uncertainty, as the review only provides an estimate for τ_{pay} but not for τ_{dep}, which is the relevant date for determining the bitcoin value of the order. The median estimated shipping time of products on Cannazon is 5 days. Therefore, $\vec{t}_{\text{dep}}$ should precede $\vec{t}_{\text{pay}}$ by no more than seven days, as the corresponding order would auto-finalize afterwards. We determine the lowest and highest bitcoin prices on the day of the review and the seven days before it and use them to estimate an interval $[\hat{\alpha}_{\text{min}}, \hat{\alpha}_{\text{max}}]$ of possible bitcoin values.

Given a specific date, Cannazon level points of a vendor are estimated by applying Equation (4) to all reviews posted prior to that date and summing up the results. We use the number of observed reviews and the count of finalized orders as specified on the vendor page to determine the fraction of customers that do not review their order and adjust our level point estimation accordingly. To account for uncertainty, we expand the estimated fee interval $[\hat{\mathrm{p}}_{\text{min}}, \hat{\mathrm{p}}_{\text{max}}]$ to include the two fee levels adjacent to our initial estimate.

We then use the definition of a matching graph as presented in Section 4.1.5 and adapt the criteria to fit Cryptonia Market and Cannazon. Consequently, we consider a payment system subgraph to be matching to a Cryptonia Market review, if (i) the graph matches the structure shown in Figure 3, (ii) $0.02 \cdot \alpha_{\text{in}} \leq \frac{\alpha^1_{\text{out}}}{\alpha_{\text{in}}} \leq 0.04 \cdot \alpha_{\text{in}}$, (iii) $\alpha \leq \alpha_{\text{in}} < \alpha + 0.0001$ and (iv) $|\tau_{\text{r}} - \tau_{\text{pay}}| \leq \delta$.

A payment system subgraph matches a Cannazon review, if (i) the graph matches the structure shown in Figure 4, (ii) $\hat{\mathrm{p}}_{\text{min}} \cdot \alpha_{\text{in}} \leq \frac{\alpha^1_{\text{out}}}{\alpha_{\text{in}}} \leq \hat{\mathrm{p}}_{\text{max}} \cdot \alpha_{\text{in}}$, (iii) $\hat{\alpha}_{\text{min}} \leq \alpha_{\text{in}} \leq \hat{\alpha}_{\text{max}}$ and (iv) $|\tau_{\text{r}} - \tau_{\text{pay}}| \leq \delta$.

In a last step we have to obtain an estimate for the review-transaction-delay δ. The high granularity of Cryptonia Market reviews allows us to identify potential vendor payout addresses based on order volume alone. This approach is substantially slower than matching on the combination of order volume and review time and also requires a larger number of reviews, making it impractical for large-scale attacks. Still, we use it to run a preliminary attack against the top 25 Cryptonia Market vendors in terms of finalized orders, excluding those accepting advance payments. We then match the payment system subgraphs observed for the 16 matched addresses to the vendor reviews closest in time and calculate the time difference in days. Of the 3,471 transaction-review pairs, 2,875 reviews were posted on the same day as the payout transaction, 256 reviews showed a time difference of one day, and 52 reviews had a difference of two days. To account for possible uncertainty resulting from different time zones of the market and the bitcoin timestamps, as well as the inaccuracy of block timestamps themselves, we allow for a difference of up to one day between review and payout transaction, resulting in an estimate of $\delta = 1$ day. This also reflects the fact that Cannazon requires vendors to manually broadcast payout transactions, leading to another potential source of delay.

The CCS threshold is set to 0.4. This value is chosen after an initial run of the experiment without the validation step. We selected a subset of five vendors that differed in terms of the orders processed, share of positive reviews and type of goods sold (physical/digital) and manually inspected their identified addresses for plausibility. The CCS was calculated for these vendors and 0.4 was determined as the rounded average of the scores.

Cryptonia Market vendors that are not identified based on address reuse are expected to have supplied BIP32 extended public keys and are retargeted with the attack on aggregation transactions. This step is skipped for Cannazon since this market does not support BIP32.

The minimum number of transaction inputs is set to four, 50% of which have to match a vendor review. These thresholds were chosen because the results from the first experiment showed that the majority of the matched vendors could be identified by two reviews.

We are aware that all thresholds are tunable parameters and that a better choice of thresholds could increase the attack performance.

6.4 Results

Using our linkage attacks, we find potential payout addresses for 308 Cryptonia Market vendors and 45 Cannazon vendors. Considering the 559 vendors who had at least two positive bitcoin-priced reviews on Cryptonia Market, this corresponds to a success rate of approximately 55.1%. For the majority of these vendors, 247 in total, the matching addresses are found during stage 1 of the attack. The second phase identifies 354 potential aggregation transactions belonging to 61 different vendor profiles. On average, these potential aggregation transactions feature approximately 10.45 inputs with a median of 5 inputs. The highest observed number of inputs is 107, the lowest one is the predefined threshold of 4. For an average of approximately 73.6% of these inputs, a matching vendor review is found. The success of our attack on Cannazon is substantially lower. Given 236 vendors having at least two positive reviews, only 19.1% of them are matched to an address. There appears to be no clear relationship between the number of reviews available for a vendor and the chance of successfully matching an address. Figure 14, Figure 15, Figure 22 and Figure 23 in the appendix provide a brief overview of the success rates for different numbers of available reviews, as well as information on the overall distribution of reviews per vendor.

6.4.1 Limitations and Discussion

It should be noted that our experimental results completely lack external validation. While the findings appear to be plausible, we cannot state with absolute certainty that the addresses found by our attack do in fact constitute addresses of Cryptonia Market or Cannazon vendors. A straightforward way of obtaining ground truth would be to make test purchases from a matched vendor and verify whether or not the payment ends up on the expected address. However, our institution's legal department strongly advised against any kind of test purchase. Law enforcement agencies and other authorities are likely bound by the same legal restrictions as we are, which further underlines the advantages of a passive attack. Therefore, we argue that the results can be validated by demonstrating the low likelihood of alternative explanations. First of all, there is a very substantial overlap between the reviews of a vendor and the incoming transactions of the matched addresses. This can be seen in Figures 16 and 17 in the appendix, which depict a vendor review page alongside a screenshot of a blockchain explorer. In addition to that, matched addresses exhibit a high transaction activity, which we would interpret as a sign of institutional ownership. Furthermore, many, if not all, transactions paying to the identified addresses match the payment system subgraphs we would expect from the respective markets. At the time of our experiment, there was a total of 33 Cannazon vendors that provided the same Pretty Good Privacy public key on their Cannazon vendor profiles as a Cryptonia Market vendor did. 16 of those vendors were successfully matched to Bitcoin addresses. After clustering the found addresses to wallets using the multi-input heuristic, we identified six vendor wallets that featured incoming transactions matching to both markets. In two cases, the attacks on Cryptonia Market and Cannazon independently matched the same payout address. In addition to that, Cannazon appears to reuse its commission addresses: We found 100 Cannazon commission addresses that received fees from transactions matched to at least two distinct vendors. Finally, we observed that the locking scripts of Cannazon multisig escrow addresses matched to the same vendor regularly include the same public key. Overall, these observations are highly unlikely to all have occurred by chance, especially given the substantial overlap between the transactions and reviews. We believe that this is evidence beyond reasonable doubt for the success of our attack.

The success rates of 55.1% and 19.1% imply that our attack cannot necessarily be used to target specific vendors. Still, the attack constitutes a substantial breach in privacy for those who have been matched. Additionally, the matched addresses can serve as a starting point from which an attacker can identify additional addresses, e. g. by recursively retrieving addresses spending to the same market commission address. Furthermore, even for the cases in which our attack does not succeed, user privacy is reduced. Since the main reason for unsuccessful matching attempts is the lack of reviews, even failed attempts result in an average candidate set of size 15 after the last iteration across both markets. For 48 unmatched vendors the final anonymity set size is five or less. We believe that manual inspection of these candidate addresses might allow to identify further vendor addresses.

The low success rate on Cannazon was to be expected, as the uncertainty with regards to the actual value of the orders was considerably higher than on Cryptonia Market. In addition to that, it is very likely that at least some of the Cannazon reviews we tried to match referred to orders that were paid for in Monero.

6.4.2 Effects of Uncertainty

We believe that price uncertainty is a primary cause of the low success rate on Cannazon. To determine whether this is the case, we re-run the attack of Cryptonia Market and provide our model with data of different granularity. We model a *time uncertainty* of u days by adjusting the definition of a matching payment system graph such that the graph is matched if $|\tau_\mathrm{r} - \tau_\mathrm{pay}| \leq u$. Similarly, *price uncertainty* is modeled by truncating the order volume stated in the reviews, α_r, to one fewer decimal place: An amount $\alpha = 0.0025$ BTC in the full information model would be truncated to $\alpha' = 0.002$ BTC. The effects of increased uncertainty can be seen in Figures 9 and 10. Due to the different approach of the aggregation transaction matching, which prevents the computation of similar metrics, the numbers refer to the address-reuse-attack only.

Figure 9 illustrates the relative success rate for the first ten iterations and different degrees of uncertainty, i. e., how many of the 247 successfully identified vendors were matched at a given iteration and uncertainty.

Increasing time uncertainty is plotted column-wise, price uncertainty row-wise, and the status quo is shown in the top-center graph. It can be seen that the attack is subject to diminishing marginal effects with respect to the number of iterations performed.

When using our base model, 126 payout addresses are already found in the second iteration. This number increases to 183 in the third and 246 in the 58^{th} iteration, before eventually converging to 247 found addresses in the 183^{rd} iteration. Interestingly, removing the time uncertainty even has a slight negative effect on the attack performance: After 88 iterations, the full information model converges to 246 matched addresses. A similar observation can be made for the case of 2-day time uncertainty, which yields 241 addresses after 32 iterations. In contrast to that, price uncertainty does indeed have an influence on both, the marginal effects as well as the overall success rate. When reducing the review amount precision to three decimal places, the count of vendor payout addresses matched by our base model drops — ceteris paribus — to 221.

Figure 10 shows the average anonymity set size during the first ten iterations for different degrees of uncertainty. Again, these numbers include the 247 vendors that were matched by our base model only. The anonymity set contains all Bitcoin addresses that match the first i observed vendor reviews, where i is the current iteration. Intuitively, the initial anonymity set size increases with additional uncertainty and decreases with the number of iterations. In the first iteration, the average anonymity set size of the highest uncertainty model (2,287.9) is two orders of magnitude larger than the one of the full information model (44.4). While the model without added uncertainty yields the smallest anonymity set size in the first iteration, allowing for two days of time uncertainty produces smallest anonymity sets from iteration four on. These findings indicate that the low success rate on Cannazon can at least be partially attributed to the increased uncertainty.

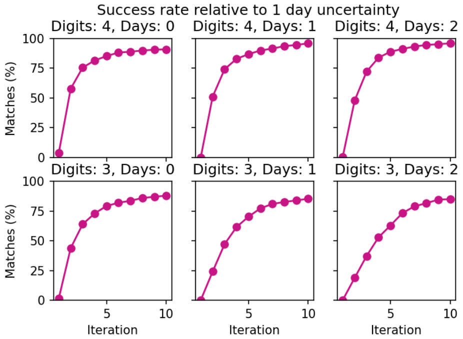


Fig. 9. Proportion of initially matched vendors re-identified under different levels of uncertainty.

6.5 Mitigation Strategies

Our findings highlight the importance of secrecy for operations security. It is acknowledged that cryptomarkets operate in an environment of distrust and uncertainty,

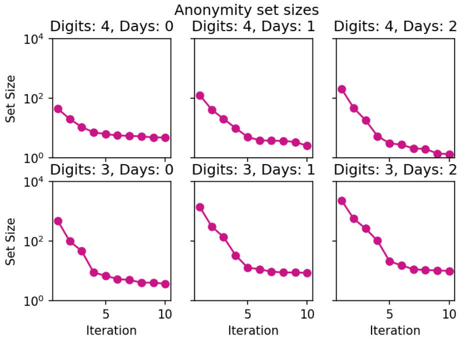


Fig. 10. Average logarithmic anonymity set size per iteration for different levels of uncertainty.

and that the detailed description of the payment system probably was an attempt to gain the confidence of potential users.

While security by obscurity is not a reliable security paradigm, releasing as little information on the payment system as possible seems to be advisable, as attacks become easier the more a-priori knowledge an attacker has. A similar case can be made for the high precision of the information embedded in the vendor reviews. As depicted in Figure 9, increases in price uncertainty have a negative impact on the overall success rate. This is also resembled in the much lower success rate of our attack on Cannazon, where we had to deal with substantially higher price uncertainty. Releasing reviews with no or very rough timestamps only would effectively prevent our attack in its current state. However, an attacker could still infer the publishing date of reviews by scraping the markets on a daily basis and comparing the found reviews to the ones from the previous day. If a review states which product was purchased, alternative approximations of the total order volume could be obtained from item prices.

Our analysis also uncovers significant weaknesses in the design of both markets' payment systems. In particular, transferring the funds directly from the escrow address to the vendor payout address results in a single point of attack, which can be easily exploited by attackers. In this regard, wallet-based centralized transfer markets (cf. Section 4.1.2) could provide higher degrees of privacy as they allow vendors to withdraw their funds in chunks of arbitrary values completely unrelated to the prices of their items. Additionally, a more privacy-preserving solution would be to channel outgoing payments through a multi-stage mixing process. Linkage attacks can be further complicated by introducing a higher degree of randomness into the payment process. Apart from the random fee sampling, which Cryptonia Market already deploys, other possible randomization strategies can be imagined: User deposits could be split across a random number of escrow addresses, leading to a random number of payout transactions. Also, a random delay could be introduced between the finalization of an order and the payout transaction, the publishing of the review, or both.

With regards to the matching of one-time payout addresses using the multi-input heuristic on aggregation transactions, the fault is less on Cryptonia Market and more on the vendors themselves. It is commonly known in the Bitcoin community that aggregation transactions may leak information on the composition of a Bitcoin wallet. Therefore, the Bitcoin community generally advises to never use different Bitcoin addresses holding funds from questionable sources as inputs for a single transaction [18].

7 Conclusion

We have introduced new attacks that allow to determine Bitcoin addresses likely belonging to darknet vendors from public information only, in particular positive reviews. Experiments confirmed the applicability of the attacks for real-world online markets.

One consequence is that online market vendors have to face two contradicting demands. On the one hand, they are interested into accumulating as many positive reviews as possible. On the other hand, exactly these reviews allow to derive information about purchases made that eventually help to track down the corresponding transactions (and hence vendor addresses).

Besides the attacks presented in this work, we consider this an interesting conflict that raises a couple of important questions for future work. First and foremost, is it possible to resolve this conflict without relying on third parties such as mixing services. Moreover, the techniques deployed in the attacks can also be used to identify user and market addresses which should be investigated further. The same holds for the question, whether an attack is still possible if timestamps, order volume or both are missing from the reviews. Last but not least, given that some decisions made in the attack were based on heuristics, higher identification rates are imaginable. Summing up, we hope that this work triggers further research on whether using cryptocurrencies is a reasonable choice for online markets, especially in the light of the currencies' rising popularity.

8 Acknowledgements

We would like to thank the reviewers for their support and valuable insights. Part of the research was carried out while the main author was employed at *SWR*, Baden-Baden, Germany. Other than that, this research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] A. Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies*. O'Reilly, Sebastopol, CA, 2017.

[2] Cannazon Market. General information. http://57iwpifn5xr7bim3lm4lywjuz45za4cbwusyerh362jiqnoraijzh2id.onion.

[3] X. Chen, M. A. Hasan, X. Wu, P. Skums, M. J. Feizollahi, M. Ouellet, E. L. Sevigny, D. Maimon, and Y. Wu. Characteristics of bitcoin transactions on cryptomarkets. In G. Wang, J. Feng, M. Z. A. Bhuiyan, and R. Lu, editors, *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, pages 261–276, Cham, 2019. Springer International Publishing.

[4] Cryptonia Market. Frequently asked questions. http://jsm5ecfs2xdjivvtizedkiuj4tgcnpewvys3qxxekvucgx2dvqxhy4qd.onion.

[5] Cryptonia Market. What are direct deposits? http://jsm5ecfs2xdjivvtizedkiuj4tgcnpewvys3qxxekvucgx2dvqxhy4qd.onion.

[6] D. Dittrich and E. Kenneally. The menlo report: Ethical principles guiding information and communication technology research. Technical report, U.S. Department of Homeland Security, 2012-08.

[7] Y. Fanusie and T. Robinson. Bitcoin laundering: an analysis of illicit flows into digital currency services. *Elliptic.co Report*, 2018.

[8] S. Goldfeder, J. Bonneau, R. Gennaro, and A. Narayanan. Escrow protocols for cryptocurrencies: How to buy physical goods using bitcoin. In A. Kiayias, editor, *Financial Cryptography and Data Security*, pages 321–339, Cham, 2017. Springer International Publishing.

[9] S. Goldfeder, H. A. Kalodner, D. Reisman, and A. Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *Proceedings on Privacy Enhancing Technologies*, 2018:179 – 199, 2017.

[10] H. Jawaheri, M. Sabah, Y. Boshmaf, and A. Erbad. Deanonymizing tor hidden service users through bitcoin transactions analysis. *Computers & Security*, 89:101684, 12 2019.

[11] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande. Characterizing entities in the bitcoin blockchain. In *Data Mining Workshop (ICDMW), 2018 IEEE International Conference on*, pages –. IEEE, 2018.

[12] N. Kshetri. Cryptocurrencies: Transparency versus privacy [cybertrust]. *Computer*, 51(11):99–111, 2018.

[13] M. Levandowsky. Distance between Sets. *Nature*, 234(5323):34–35, Nov. 1971.

[14] D. McGinn, D. McIlwraith, and Y. Guo. Toward open data blockchain analytics: A bitcoin perspective. *Royal Society Open Science*, 5, 02 2018.

[15] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 127–140, New York, NY, USA, 2013. ACM.

[16] F. Sabry, W. Labda, A. Erbad, H. Al Jawaheri, and Q. Malluhi. Anonymity and privacy in bitcoin escrow trades. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pages 211–220, 2019.

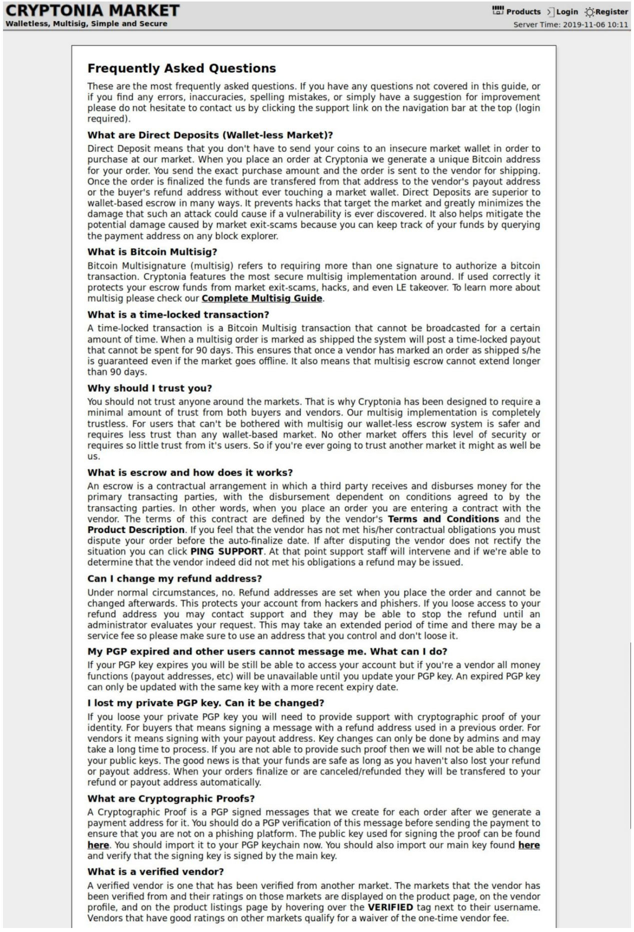
[17] D. Sommer. Processing bitcoin blockchain data using a big data-specific framework. Bachelor's Thesis, University of Zurich. https://www.merlin.uzh.ch/contributionDocument/download/11801, 05 2019.

[18] The Bitcoin Wiki contributors. Privacy, section 9.10. http://archive.today/qY7of, 06 2019.

[19] P. Wuille. Bitcoin improvement protocol 32, 02 2012.

[20] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren. Bitcoin and blockchain: Security and privacy. *IEEE Internet of Things Journal*, 7(10):10288–10313, 2020.

A Appendix


CRYPTONIA MARKET
Walletless, Multisig, Simple and Secure
Products
Login
Register
Server Time: 2019-11-06 10:11

Frequently Asked Questions

These are the most frequently asked questions. If you have any questions not covered in this guide, or if you find any errors, inaccuracies, spelling mistakes, or simply have a suggestion for improvement please do not hesitate to contact us by clicking the support link on the navigation bar at the top (login required).

What are Direct Deposits (Wallet-less Market)?

Direct Deposit means that you don't have to send your coins to an insecure market wallet in order to purchase at our market. When you place an order at Cryptonia we generate a unique Bitcoin address for your order. You send the exact purchase amount and the order is sent to the vendor for shipping. Once the order is finalized the funds are transfered from that address to the vendor's payout address or the buyer's refund address without ever touching a market wallet. Direct Deposits are superior to wallet-based escrow in many ways. It prevents hacks that target the market and greatly minimizes the damage that such an attack could cause if a vulnerability is ever discovered. It also helps mitigate the potential damage caused by market exit-scams because you can keep track of your funds by querying the payment address on any block explorer.

What is Bitcoin Multisig?

Bitcoin Multisignature (multisig) refers to requiring more than one signature to authorize a bitcoin transaction. Cryptonia features the most secure multisig implementation around. If used correctly it protects your escrow funds from market exit-scams, hacks, and even LE takeover. To learn more about multisig please check our **Complete Multisig Guide**.

What is a time-locked transaction?

A time-locked transaction is a Bitcoin Multisig transaction that cannot be broadcasted for a certain amount of time. When a multisig order is marked as shipped the system will post a time-locked payout that cannot be spent for 90 days. This ensures that once a vendor has marked an order as shipped s/he is guaranteed even if the market goes offline. It also means that multisig escrow cannot extend longer than 90 days.

Why should I trust you?

You should not trust anyone around the markets. That is why Cryptonia has been designed to require a minimal amount of trust from both buyers and vendors. Our multisig implementation is completely trustless. For users that can't be bothered with multisig our wallet-less escrow system is safer and requires less trust than any wallet-based market. No other market offers this level of security or requires so little trust from it's users. So if you're ever going to trust another market it might as well be us.

What is escrow and how does it works?

An escrow is a contractual arrangement in which a third party receives and disburses money for the primary transacting parties, with the disbursement dependent on conditions agreed to by the transacting parties. In other words, when you place an order you are entering a contract with the vendor. The terms of this contract are defined by the vendor's **Terms and Conditions** and the **Product Description**. If you feel that the vendor has not met his/her contractual obligations you must dispute your order before the auto-finalize date. If after disputing the vendor does not rectify the situation you can click **PING SUPPORT**. At that point support staff will intervene and if we're able to determine that the vendor indeed did not met his obligations a refund may be issued.

Can I change my refund address?

Under normal circumstances, no. Refund addresses are set when you place the order and cannot be changed afterwards. This protects your account from hackers and phishers. If you loose access to your refund address you may contact support and they may be able to stop the refund until an administrator evaluates your request. This may take an extended period of time and there may be a service fee so please make sure to use an address that you control and don't loose it.

My PGP expired and other users cannot message me. What can I do?

If your PGP key expires you will be still be able to access your account but if you're a vendor all money functions (payout addresses, etc) will be unavailable until you update your PGP key. An expired PGP key can only be updated with the same key with a more recent expiry date.

I lost my private PGP key. Can it be changed?

If you loose your private PGP key you will need to provide support with cryptographic proof of your identity. For buyers that means signing a message with a refund address used in a previous order. For vendors it means signing with your payout address. Key changes can only be done by admins and may take a long time to process. If you are not able to provide such proof then we will not be able to change your public keys. The good news is that your funds are safe as long as you haven't also lost your refund or payout address. When your orders finalize or are canceled/refunded they will be transfered to your refund or payout address automatically.

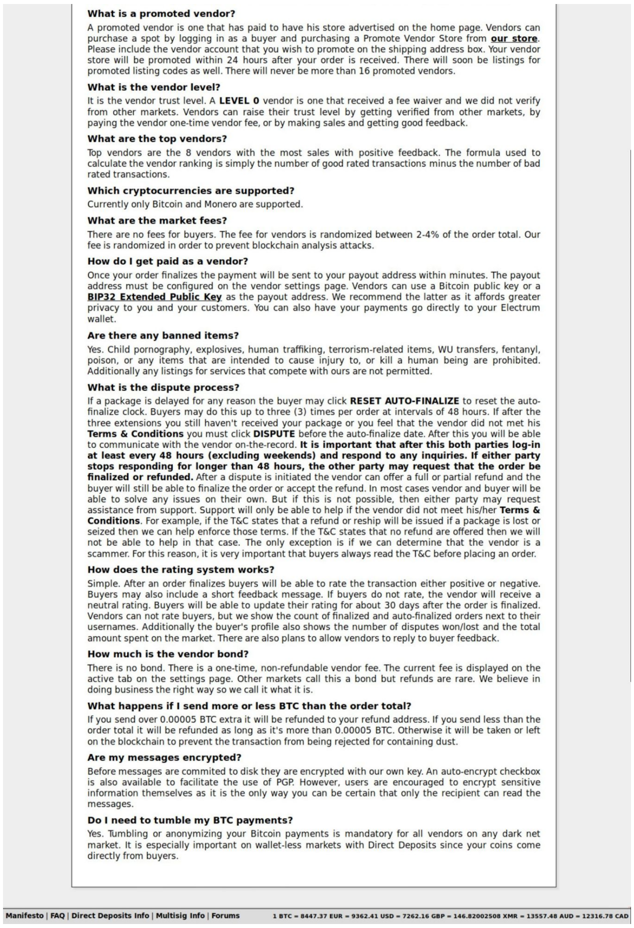
What are Cryptographic Proofs?

A Cryptographic Proof is a PGP signed messages that we create for each order after we generate a payment address for it. You should do a PGP verification of this message before sending the payment to ensure that you are not on a phishing platform. The public key used for signing the proof can be found **here**. You should import it to your PGP keychain now. You should also import our main key found **here** and verify that the signing key is signed by the main key.

What is a verified vendor?

A verified vendor is one that has been verified from another market. The markets that the vendor has been verified from and their ratings on those markets are displayed on the product page, on the vendor profile, and on the product listings page by hovering over the **VERIFIED** tag next to their username. Vendors that have good ratings on other markets qualify for a waiver of the one-time vendor fee.

Fig. 11. Cryptonia's payment info page 1/2.



What is a promoted vendor?

A promoted vendor is one that has paid to have his store advertised on the home page. Vendors can purchase a spot by logging in as a buyer and purchasing a Promote Vendor Store from **our store**. Please include the vendor account that you wish to promote on the shipping address box. Your vendor store will be promoted within 24 hours after your order is received. There will soon be listings for promoted listing codes as well. There will never be more than 16 promoted vendors.

What is the vendor level?

It is the vendor trust level. A **LEVEL 0** vendor is one that received a fee waiver and we did not verify from other markets. Vendors can raise their trust level by getting verified from other markets, by paying the vendor one-time vendor fee, or by making sales and getting good feedback.

What are the top vendors?

Top vendors are the 8 vendors with the most sales with positive feedback. The formula used to calculate the vendor ranking is simply the number of good rated transactions minus the number of bad rated transactions.

Which cryptocurrencies are supported?

Currently only Bitcoin and Monero are supported.

What are the market fees?

There are no fees for buyers. The fee for vendors is randomized between 2-4% of the order total. Our fee is randomized in order to prevent blockchain analysis attacks.

How do I get paid as a vendor?

Once your order finalizes the payment will be sent to your payout address within minutes. The payout address must be configured on the vendor settings page. Vendors can use a Bitcoin public key or a **BIP32 Extended Public Key** as the payout address. We recommend the latter as it affords greater privacy to you and your customers. You can also have your payments go directly to your Electrum wallet.

Are there any banned items?

Yes. Child pornography, explosives, human traffiking, terrorism-related items, WU transfers, fentanyl, poison, or any items that are intended to cause injury to, or kill a human being are prohibited. Additionally any listings for services that compete with ours are not permitted.

What is the dispute process?

If a package is delayed for any reason the buyer may click **RESET AUTO-FINALIZE** to reset the auto-finalize clock. Buyers may do this up to three (3) times per order at intervals of 48 hours. If after the three extensions you still haven't received your package or you feel that the vendor did not met his **Terms & Conditions** you must click **DISPUTE** before the auto-finalize date. After this you will be able to communicate with the vendor on-the-record. **It is important that after this both parties log-in at least every 48 hours (excluding weekends) and respond to any inquiries. If either party stops responding for longer than 48 hours, the other party may request that the order be finalized or refunded.** After a dispute is initiated the vendor can offer a full or partial refund and the buyer will still be able to finalize the order or accept the refund. In most cases vendor and buyer will be able to solve any issues on their own. But if this is not possible, then either party may request assistance from support. Support will only be able to help if the vendor did not meet his/her **Terms & Conditions**. For example, if the T&C states that a refund or reship will be issued if a package is lost or seized then we can help enforce those terms. If the T&C states that no refund are offered then we will not be able to help in that case. The only exception is if we can determine that the vendor is a scammer. For this reason, it is very important that buyers always read the T&C before placing an order.

How does the rating system works?

Simple. After an order finalizes buyers will be able to rate the transaction either positive or negative. Buyers may also include a short feedback message. If buyers do not rate, the vendor will receive a neutral rating. Buyers will be able to update their rating for about 30 days after the order is finalized. Vendors can not rate buyers, but we show the count of finalized and auto-finalized orders next to their usernames. Additionally the buyer's profile also shows the number of disputes won/lost and the total amount spent on the market. There are also plans to allow vendors to reply to buyer feedback.

How much is the vendor bond?

There is no bond. There is a one-time, non-refundable vendor fee. The current fee is displayed on the active tab on the settings page. Other markets call this a bond but refunds are rare. We believe in doing business the right way so we call it what it is.

What happens if I send more or less BTC than the order total?

If you send over 0.00005 BTC extra it will be refunded to your refund address. If you send less than the order total it will be refunded as long as it's more than 0.00005 BTC. Otherwise it will be taken or left on the blockchain to prevent the transaction from being rejected for containing dust.

Are my messages encrypted?

Before messages are commited to disk they are encrypted with our own key. An auto-encrypt checkbox is also available to facilitate the use of PGP. However, users are encouraged to encrypt sensitive information themselves as it is the only way you can be certain that only the recipient can read the messages.

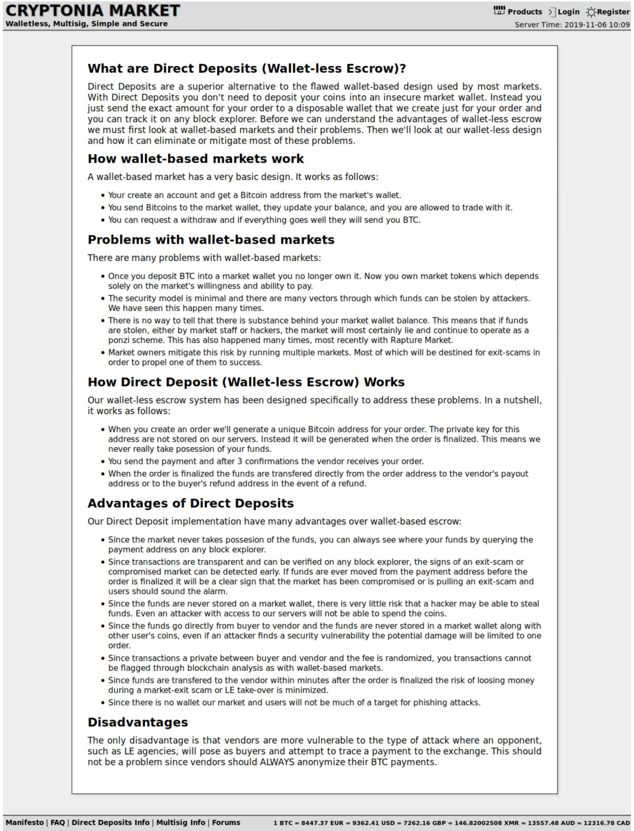
Do I need to tumble my BTC payments?

Yes. Tumbling or anonymizing your Bitcoin payments is mandatory for all vendors on any dark net market. It is especially important on wallet-less markets with Direct Deposits since your coins come directly from buyers.

Manifesto | FAQ | Direct Deposits Info | Multisig Info | Forums

1 BTC = 8447.37 EUR = 9362.41 USD = 7262.16 GBP = 146.82002508 XMR = 13557.48 AUD = 12316.78 CAD

Fig. 12. Cryptonia's payment info page 2/2.



CRYPTONIA MARKET
Walletless, Multisig, Simple and Secure

Server Time: 2019-11-06 10:09

What are Direct Deposits (Wallet-less Escrow)?

Direct Deposits are a superior alternative to the flawed wallet-based design used by most markets. With Direct Deposits you don't need to deposit your coins into an insecure market wallet. Instead you just send the exact amount for your order to a disposable wallet that we create just for your order and you can track it on any block explorer. Before we can understand the advantages of wallet-less escrow we must first look at wallet-based markets and their problems. Then we'll look at our wallet-less design and how it can eliminate or mitigate most of these problems.

How wallet-based markets work

A wallet-based market has a very basic design. It works as follows:

- Your create an account and get a Bitcoin address from the market's wallet.
- You send Bitcoins to the market wallet, they update your balance, and you are allowed to trade with it.
- You can request a withdraw and if everything goes well they will send you BTC.

Problems with wallet-based markets

There are many problems with wallet-based markets:

- Once you deposit BTC into a market wallet you no longer own it. Now you own market tokens which depends solely on the market's willingness and ability to pay.
- The security model is minimal and there are many vectors through which funds can be stolen by attackers. We have seen this happen many times.
- There is no way to tell that there is substance behind your market wallet balance. This means that if funds are stolen, either by market staff or hackers, the market will most certainly lie and continue to operate as a ponzi scheme. This has also happened many times, most recently with Rapture Market.
- Market owners mitigate this risk by running multiple markets. Most of which will be destined for exit-scams in order to propel one of them to success.

How Direct Deposit (Wallet-less Escrow) Works

Our wallet-less escrow system has been designed specifically to address these problems. In a nutshell, it works as follows:

- When you create an order we'll generate a unique Bitcoin address for your order. The private key for this address are not stored on our servers. Instead it will be generated when the order is finalized. This means we never really take posession of your funds.
- You send the payment and after 3 confirmations the vendor receives your order.
- When the order is finalized the funds are transfered directly from the order address to the vendor's payout address or to the buyer's refund address in the event of a refund.

Advantages of Direct Deposits

Our Direct Deposit implementation have many advantages over wallet-based escrow:

- Since the market never takes possesion of the funds, you can always see where your funds by querying the payment address on any block explorer.
- Since transactions are transparent and can be verified on any block explorer, the signs of an exit-scam or compromised market can be detected early. If funds are ever moved from the payment address before the order is finalized it will be a clear sign that the market has been compromised or is pulling an exit-scam and users should sound the alarm.
- Since the funds are never stored on a market wallet, there is very little risk that a hacker may be able to steal funds. Even an attacker with access to our servers will not be able to spend the coins.
- Since the funds go directly from buyer to vendor and the funds are never stored in a market wallet along with other user's coins, even if an attacker finds a security vulnerability the potential damage will be limited to one order.
- Since transactions a private between buyer and vendor and the fee is randomized, you transactions cannot be flagged through blockchain analysis as with wallet-based markets.
- Since funds are transfered to the vendor within minutes after the order is finalized the risk of loosing money during a market-exit scam or LE take-over is minimized.
- Since there is no wallet our market and users will not be much of a target for phishing attacks.

Disadvantages

The only disadvantage is that vendors are more vulnerable to the type of attack where an opponent, such as LE agencies, will pose as buyers and attempt to trace a payment to the exchange. This should not be a problem since vendors should ALWAYS anonymize their BTC payments.

Manifesto | FAQ | Direct Deposits Info | Multisig Info | Forums

1 BTC = 8447.37 EUR = 9362.41 USD = 7262.16 GBP = 146.82002508 XMR = 13557.48 AUD = 12316.78 CAD

Fig. 13. Cryptonia's FAQ page.

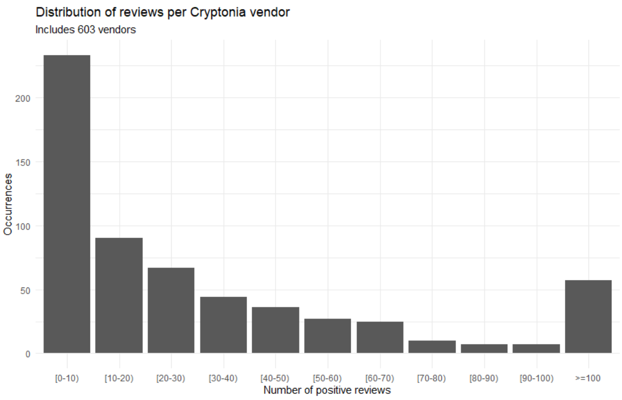


Fig. 14. Distribution of reviews per Cryptonia Market vendor.

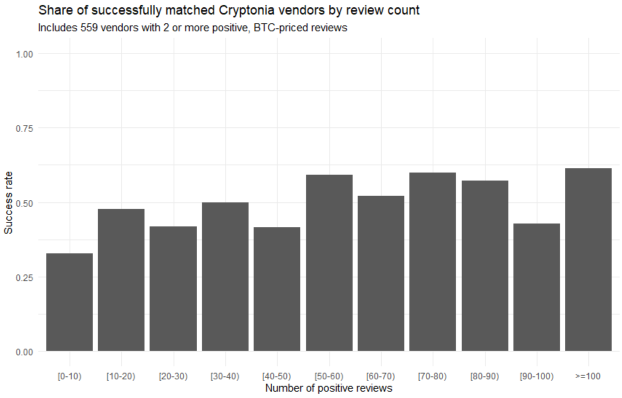
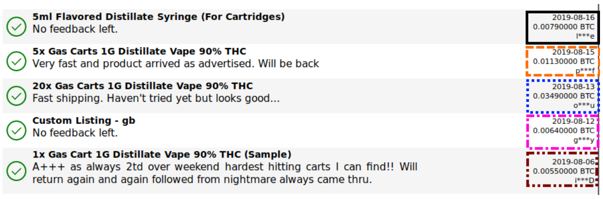
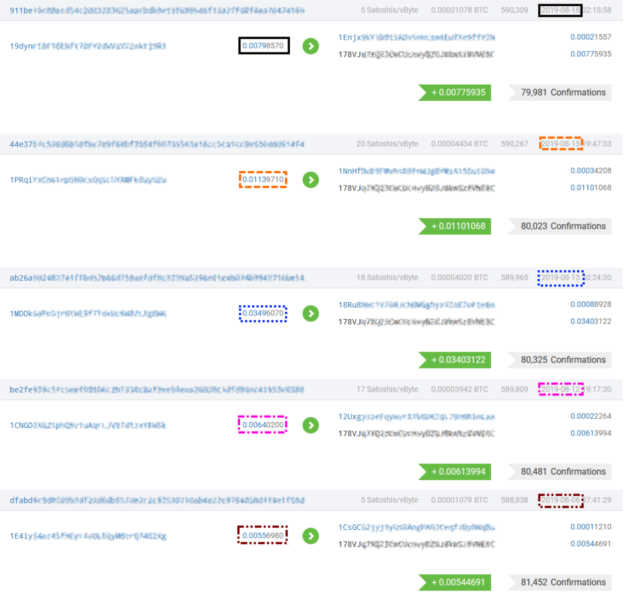


Fig. 15. Success rate by review count on Cryptonia Market.



✓	**5ml Flavored Distillate Syringe (For Cartridges)** No feedback left.	2019-08-16 0.00790000 BTC l***e
✓	**5x Gas Carts 1G Distillate Vape 90% THC** Very fast and product arrived as advertised. Will be back	2019-08-15 0.01130000 BTC p***f
✓	**20x Gas Carts 1G Distillate Vape 90% THC** Fast shipping. Haven't tried yet but looks good...	2019-08-13 0.03490000 BTC o***u
✓	**Custom Listing - gb** No feedback left.	2019-08-12 0.00640000 BTC g***y
✓	**1x Gas Cart 1G Distillate Vape 90% THC (Sample)** A+++ as always 2td over weekend hardest hitting carts I can find!! Will return again and again followed from nightmare always came thru.	2019-08-06 0.00550000 BTC i***D

Fig. 16. Review page of a Cryptonia Market vendor (cropped).



911be[redacted] 5 Satoshis/vByte 0.00001078 BTC 590,309 2019-08-16 02:15:58
19dyn[redacted] 0.00798570 → 1Enjx[redacted] 0.00021557
178VJ[redacted] 0.00775935
+ 0.00775935 79,981 Confirmations

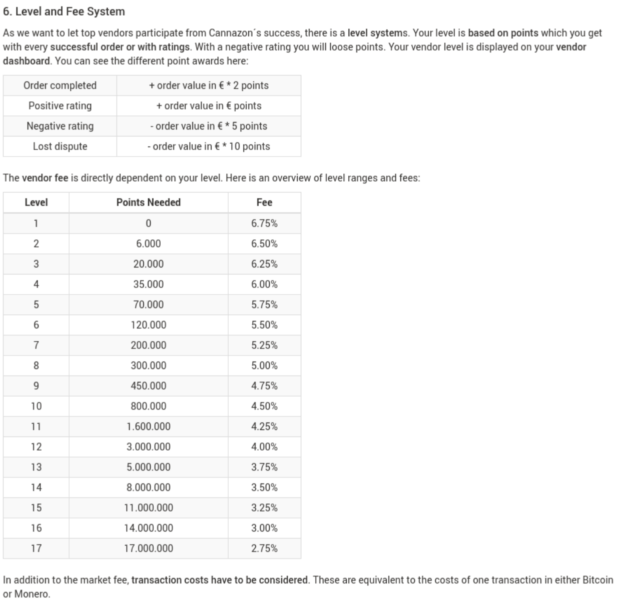
44e37[redacted] 20 Satoshis/vByte 0.00004434 BTC 590,267 2019-08-15 19:47:33
1PRqi[redacted] 0.01139710 → 1NnHf[redacted] 0.00034208
178VJ[redacted] 0.01101068
+ 0.01101068 80,023 Confirmations

ab26a[redacted] 18 Satoshis/vByte 0.00004020 BTC 589,965 2019-08-13 20:24:30
1MDDk[redacted] 0.03496070 → 18Ru8[redacted] 0.00088928
178VJ[redacted] 0.03403122
+ 0.03403122 80,325 Confirmations

be2fe[redacted] 17 Satoshis/vByte 0.00003942 BTC 589,809 2019-08-12 19:17:30
1CNGD[redacted] 0.00640200 → 12Uxg[redacted] 0.00022264
178VJ[redacted] 0.00613994
+ 0.00613994 80,481 Confirmations

dfabd[redacted] 5 Satoshis/vByte 0.00001079 BTC 588,838 2019-08-06 07:41:29
1E4iy[redacted] 0.00556980 → 1CsGC[redacted] 0.00011210
178VJ[redacted] 0.00544691
+ 0.00544691 81,452 Confirmations

Fig. 17. Matched payout address on btc.com (cropped and edited). Matches of reviews and transactions are marked with matching rectangles.



6. Level and Fee System

As we want to let top vendors participate from Cannazon´s success, there is a **level systems**. Your level is **based on points** which you get with every **successful order or with ratings**. With a negative rating you will loose points. Your vendor level is displayed on your **vendor dashboard**. You can see the different point awards here:

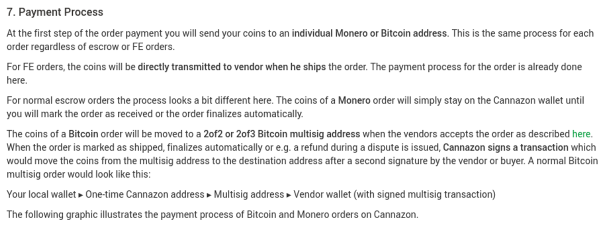
Order completed	+ order value in € * 2 points
Positive rating	+ order value in € points
Negative rating	- order value in € * 5 points
Lost dispute	- order value in € * 10 points

The **vendor fee** is directly dependent on your level. Here is an overview of level ranges and fees:

Level	Points Needed	Fee
1	0	6.75%
2	6.000	6.50%
3	20.000	6.25%
4	35.000	6.00%
5	70.000	5.75%
6	120.000	5.50%
7	200.000	5.25%
8	300.000	5.00%
9	450.000	4.75%
10	800.000	4.50%
11	1.600.000	4.25%
12	3.000.000	4.00%
13	5.000.000	3.75%
14	8.000.000	3.50%
15	11.000.000	3.25%
16	14.000.000	3.00%
17	17.000.000	2.75%

In addition to the market fee, **transaction costs have to be considered**. These are equivalent to the costs of one transaction in either Bitcoin or Monero.

Fig. 18. Calculation of vendor fees as shown on Cannazon.



7. Payment Process

At the first step of the order payment you will send your coins to an **individual Monero or Bitcoin address**. This is the same process for each order regardless of escrow or FE orders.

For FE orders, the coins will be **directly transmitted to vendor when he ships** the order. The payment process for the order is already done here.

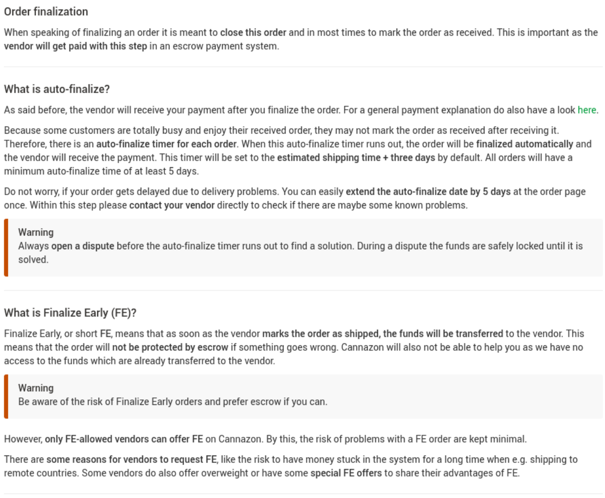
For normal escrow orders the process looks a bit different here. The coins of a **Monero** order will simply stay on the Cannazon wallet until you will mark the order as received or the order finalizes automatically.

The coins of a **Bitcoin** order will be moved to a **2of2 or 2of3 Bitcoin multisig address** when the vendors accepts the order as described here. When the order is marked as shipped, finalizes automatically or e.g. a refund during a dispute is issued, **Cannazon signs a transaction** which would move the coins from the multisig address to the destination address after a second signature by the vendor or buyer. A normal Bitcoin multisig order would look like this:

Your local wallet ▸ One-time Cannazon address ▸ Multisig address ▸ Vendor wallet (with signed multisig transaction)

The following graphic illustrates the payment process of Bitcoin and Monero orders on Cannazon.

Fig. 19. Cannazon's description of the payment process.



Order finalization

When speaking of finalizing an order it is meant to **close this order** and in most times to mark the order as received. This is important as the **vendor will get paid with this step** in an escrow payment system.

What is auto-finalize?

As said before, the vendor will receive your payment after you finalize the order. For a general payment explanation do also have a look here.

Because some customers are totally busy and enjoy their received order, they may not mark the order as received after receiving it. Therefore, there is an **auto-finalize timer for each order**. When this auto-finalize timer runs out, the order will be **finalized automatically** and the vendor will receive the payment. This timer will be set to the **estimated shipping time + three days** by default. All orders will have a minimum auto-finalize time of at least 5 days.

Do not worry, if your order gets delayed due to delivery problems. You can easily **extend the auto-finalize date by 5 days** at the order page once. Within this step please **contact your vendor** directly to check if there are maybe some known problems.

> Warning
> Always **open a dispute** before the auto-finalize timer runs out to find a solution. During a dispute the funds are safely locked until it is solved.

What is Finalize Early (FE)?

Finalize Early, or short **FE**, means that as soon as the vendor **marks the order as shipped, the funds will be transferred** to the vendor. This means that the order will **not be protected by escrow** if something goes wrong. Cannazon will also not be able to help you as we have no access to the funds which are already transferred to the vendor.

> Warning
> Be aware of the risk of Finalize Early orders and prefer escrow if you can.

However, **only FE-allowed vendors can offer FE** on Cannazon. By this, the risk of problems with a FE order are kept minimal.

There are **some reasons for vendors to request FE**, like the risk to have money stuck in the system for a long time when e.g. shipping to remote countries. Some vendors do also offer overweight or have some **special FE offers** to share their advantages of FE.

Fig. 20. Cannazon's explanation of the escrow system.

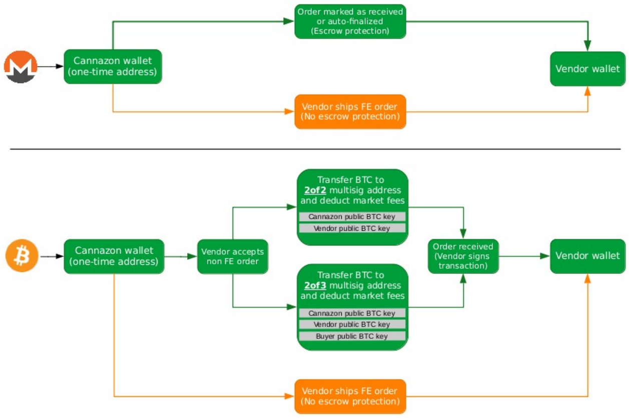


Fig. 21. Structure of the payment process as shown on Cannazon.

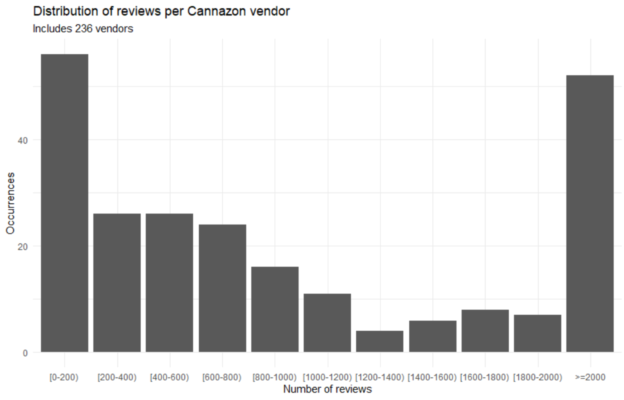


Fig. 22. Distribution of reviews per Cannazon vendor.

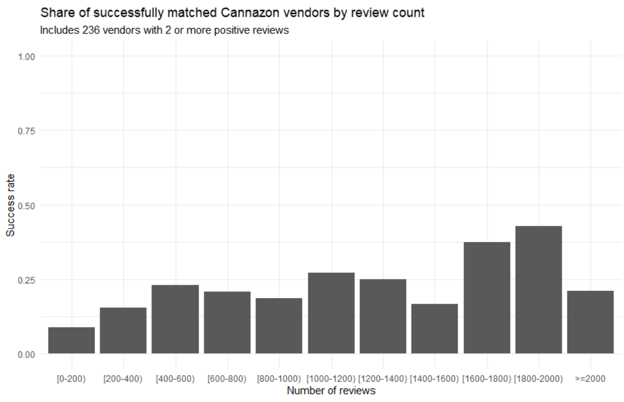


Fig. 23. Success rate by review count on Cannazon.