Adversarial behaviours in mixing coins under incomplete information

Accepted refereed manuscript of: Wang Y, Bracciali A, Yang G, Li T & Yu X (2020) Adversarial behaviours in mixing coins under incomplete information. *Applied Soft Computing*, 96, Art. No.: 106605. https://doi.org/10.1016/j.asoc.2020.106605

Adversarial Behaviours in Mixing Coins under Incomplete Information

Yilei Wang*[a,b], Andrea Bracciali[c], Guoyu Yang[a], Tao Li[a], Fengyin Li[a,d], Xiaomei Yu[e]

[a] *School of Information Science and Engineering, Qufu Normal University, China*
[b] *Institute of Artificial Intelligence and Blockchain, Guangzhou University, China*
[c] *Computing Science and Mathematics, University of Stirling, UK*
[d] *School of Informatics, University of Leicester, UK*
[e] *School of Information Science and Engineering, Shandong Normal University, China*

Abstract

Criminals can launder crypto-currencies through mixing coins, whose original purpose is preservation of privacy in the presence of traceability. Therefore, it's essential to elaborately design mixing polices to achieve both privacy and anti-money laundering. Existing work on mixing policies relies on the knowledge of a blacklist. However, these policies are paralyzed under the scenario where the blacklist is unknown or evolving. In this paper, we regard the above scenario as games under incomplete information where parties put down a deposit for the quality of coins, which is suitably managed by a smart contract in case of mixing bad coins. We extend the *poison* and *haircut* policies to incomplete information games, where the blacklist is updated after mixing. We prove the existence of equilibria for the improved polices, while it is known that there is no equilibria in the original *poison* and *haircut* policies, where blacklist is public known. Furthermore, we propose a seminal *suicide* policy: the one who mixes more bad coins will be punished by not having the deposit refunded. Thus, parties have no incentives to launder money by leveraging mixing coins. In effect, all three policies contrast money laundering while preserving privacy under incomplete information. Finally, we simulate and verify the validity of

Email addresses: ylwangqfnu@163.com (Yilei Wang*), abracciali@gmail.com (Andrea Bracciali), yangguoyu1020@163.com (Guoyu Yang), litao_ldu@163.com (Tao Li), lfyin318@126.com (Fengyin Li), yxm0708@126.com (Xiaomei Yu)

these policies.

Keywords: Mixing coins, Incomplete information, Smart contract, Equilibrium

1. Introduction

The whole transactions' history in blockchain is transparent by resorting to some probing tools. For instance, it's easy to trace the source of bitcoins even if the accounts are anonymous. Some empirical experiments indicate that anonymous parties of bitcoins in the cyberspace can be traced to their true identities in real world. As a consequence, parties' privacy is sabotaged without protection for identities and transactions. The seminal idea of mixing coins consists of collecting coins from several sources and distributing coins to various targets. The basic principle is to strip the addressed of source and targets such that no traceability can be implemented by probing tools. Thus the privacy and anonymity are strengthened.

The most popular mixer is based on coin mixing protocols without host. e.g Coinjoin and Coinshuffle [1, 2, 3]. Parties can mix their coins through certain service providers, such as Wasabi wallet and Samourai wallet. However it leaves mixing coins vulnerable to adversarial behaviours, such as money laundry. For example, in December 2019, one account has been suspended by Binance Singapore since it attempted to send bitcoins to Wasabi wallet. Binance announced that the account violated Binance Singapore's anti-money laundering policy and triggered a risk alert from the monetary authority of Singapore. In February 2020, US authorities charged DropBit CEO for allegedly laundering bitcoins for 311M USD. The indictment asserts that the Helix mixer is suspected of being involved in remittances and money laundering.[1]

Therefore, new mechanism of mixing coins are clearly needed to improve the tradeoff between privacy and adversarial purposes. One can make a tradeoff

[1] https://www.bleepingcomputer.com/news/security/helix-bitcoin-mixer-owner-charged-for-laundering-over-310-million/

by utilizing deposits managed by smart contracts according to game theory principles [4, 5]. Note that we may also utilize some advanced technology like machine learning [6], data mining [7] and secure multi-party computation [8, 9] to solve these kind of problems since adversaries need to learn the quality of bad coins. Here the quality of bad coins means the ratio of the bad coins.

In this paper our results are established on the assumption of a blacklist with adversarial learning, and therefore we will focus on adversarial behaviours through the view of game theory under incomplete information [10, 11], which is a more complex scenario for machine learning system. We assume the existence of a blacklist dynamically updated as a service managed by a trusted third party. The initial listing of a coin in the blacklist is clearly critical and requires trust in the blacklist manager. Information about transactions, on which the blacklist is based, is actually public and verifiable and does not require trust. Other judgements may also be easily verifiable, consider for instance coins from a publicly known ransomware address. However, blacklist management may clearly constitute a dominant position. The problem of trust in blacklists is scope for future work.

1.1. Related work

There have been a flurry of related works in mixing coins, while this work sets out to address the problem of privacy preserving and anti-money launder. Many empirical works focus on other problems such as phishing protection [12] and data sharing [13]. However, these works are not fully convincing, not being suitably supported by formal analysis. Edward et al. explore the model for malware infections in blacklisting by utilizing a simple Markov model [14]. Hofmeyr et al. trade off between prevented harm and collateral damage by modelling potential policy interventions [15]. In fact, it's better to describe these problems in game theory with economic views. For example, Acquisti et al. [16] discuss the economic incentives in message anonymization from the viewpoint of game theory, even if blockchains had not been defined at the time.

Moser et al. propose three policies to assess risk in bitcoin transactions,

focusing on risk scoring and the implications of blacklisting: *poison, haircut* and *seniority* for quality propagation [17]. Bonneau et al. [18] borrow the above three policies and analyse them toward game-theoretic view. More specifically, they formalize the game of mixing coins with different quality. They analyse the conditions to reach equilibria for different scenarios including perfect games and imperfect simultaneous-move games. However, their work does not consider extensive games under incomplete information, which is more suitable for real application scenarios[2].

1.2. Motivations and contributions

The original incentive for mixing coins is to provide privacy for sponsors (aka. privacy seeker), who pay privacy providers for the service. However, perpetrators may launder money by mixing with bad coins, which jeopardize the original incentives for mixing coins. Indeed, it is a better practice to only mix coins with good coins. That is, parties with bad coins should bear no incentives to mixing coin services.

The intuition of the approach followed in this paper is that privacy seekers and mixing providers commit deposits through smart contracts before they mix coins. The ones who attempt to mix bad coins will be punished by deducting their deposits. Thus, perpetrators have no incentives to mix bad coins especially when the deposits are larger than bad coins. Such a trivial solution can be trivially achieved by implementing deposit mechanisms in the *poison* and *haircut* policy of [22]. Unfortunately, these policies are not geared to the following scenario: the blacklist is updated during the process of mixing coins and the parties therein are allowed to alternatively take actions. Although the work of [22] discusses the scenario under imperfect information, they only analyze simultaneous games, where parties take actions at the same time.

In this paper, we consider a more complex scenario, i.e. extensive games

[2]Note that fuzzy theory can be also used to solve a number of relevant problems. However, this is out of scope here. Interested readers may refer to [19, 20, 21] for more details.

under incomplete information. In Figure 1, the blacklist is updated within an interval T and the time for mixing coins is $S > T$. Initially, the blacklist L is partial knowledge to Bob since bad coins x are not updated to the blacklist. Bob deems x as good coins and mixes his coins y with Alice's coins x. The dotted boxes in mixing transaction tr_{old} denote the outputs in the context that x are good coins. However, the blacklist is updated to L' after interval T, and x are detected as bad coins in L'. Therefore, the receivers get outputs in tr_{new} instead of tr_{old}. The gray boxes denote the outputs with bad coins[3]. Consequently, Alice successfully mixes bad coins under such scenario. In the following of this paper, incomplete information refer to scenarios similar to the one in Figure 1, if not differently specified.

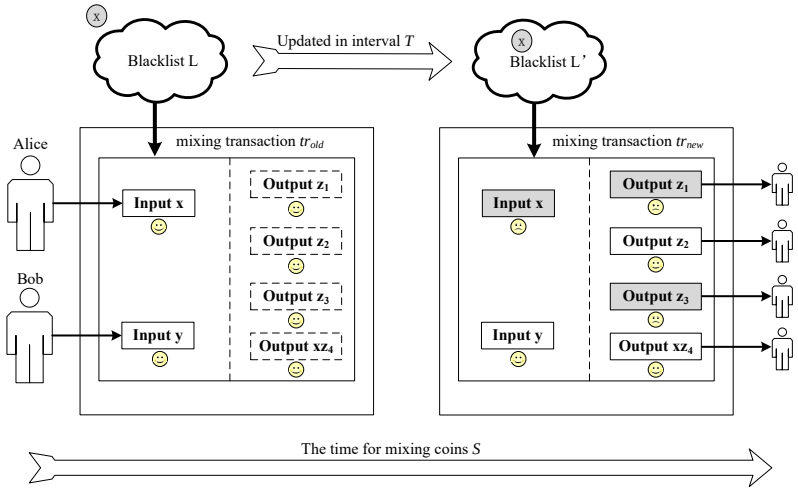

Figure 1: The mixing coins under incomplete information.

We revisit *poison* and *haircut* policies and extend them to incomplete information; we discuss the limits of *seniority* policy in the presence of incomplete information; and we overcome such limitations by proposing the new *suicide*

[3]Note that we do not demonstrate the concrete policy here. Therefore, the gray boxes just symbols for the outputs with bad coins and are not generated by any specific policy.

policy, which enforces the punishment to those that mix bad coins. The main contributions of this paper are as follows:

- *Poison* and *haircut* policy are revisited for mixing coins and extended to incomplete information. A deposit mechanism based on a smart contract is introduced to facilitate privacy protecting while avoiding money laundering trough mixing coins.

- We show that *seniority* policy is invalid under incomplete information, where a blacklist is not timely updated. This is due, intuitively speaking, to the fact that the policy fails to negotiate the order and amounts of transaction's outputs since the blacklist does not necessarily include all bad coins at the beginning of mixing.

- We propose *suicide* policy, as an improved *seniority* policy, to punish the parties with a low-level of good coins. That is, all bad coins are assigned to the parties who are mixing with the larger numbers of bad coins. Furthermore, deposits are not refund to the one who mix with the larger numbers of bad coins.

- We prove the existence of equilibria for the *poison, haircut* and *suicide* policy, respectively. We carry out simulations, whose results show the validity of these new policies under incomplete information. Furthermore, privacy seekers and providers may choose optimal mix strategies to reach a trade-off between privacy and anti-money laundering.

1.3. Road map

The remainder of this paper is organized as follows. Section 2 presents some preliminaries, e.g. different types of games and mixing coins. An extensive game under incomplete information is described in Section 3, where we bridge the notions between game theory and our model, and redefine utilities for our model. We analyze the equilibria conditions for each policy in Section 4. Moreover, we propose a new policy to solve the problem in *seniority*, where two parties cannot

negotiate under incomplete information. In section 5, simulations are shown to empirically and visually support out theoretical analysis. Section 6 summarizes results and looks into future work.

2. Background on games and mixing policies

2.1. Games under incomplete information

In game theory, parties are assumed to be rational and aim to to maximize their utilities. Games fall into various categories according to different rules. Static and extensive games are defined according to whether parities take actions simultaneously; perfect and imperfect games are defined according to whether parties achieve complete action information with respect to their opponents; complete and incomplete games are defined whether parties may or may not learn all information about the game. These games can arbitrarily be combined in new categories, like static games under perfect information, incomplete games under incomplete information, etc. In this paper, we focus on *extensive games under incomplete information*, where parties take actions alternatively and they observe the actions of forgoers. Incomplete information here means that parties have partial information on strategy space, utilities etc. More specifically, parties only have partial information about the blacklist when they make decisions: due to the lack of timely updates, the blacklist becomes partial information for parties. However, for simplicity, we assume that the blacklist will be updated during the game, and it is common knowledge for both parties. We rely on smart contracts that may decide whether to refund deposits or not according to the blacklist.

For completeness, we repeat the definition of extensive game under incomplete information as in [23].

Definition 1. An extensive game with incomplete information

$$< \mathcal{P}, A_{i\in\{1,2..\}}, H, P, f_c, (\mathcal{I}_{P_i})_{i\in\{1,2..\}}, U_{i\in\{1,2..\}} >$$

is described as follows.

1. A set of parties $\mathcal{P} = \{P_i\}$, $i \in \{1, 2, ...\}$.
2. A set of actions A_i for each P_i. The action profile is denoted as $a = (a_1, a_2, ..., ...)$, where $a_i \in A_i$.
3. A set of histories H consisting of sequences actions of assigned parties. The set of actions available after the nonterminal histories is denoted by Z.
4. A function P that assigns to each nonterminal history a party $P_j \in \mathcal{P}$.
5. For each party P_j, an information set is denoted by $\mathcal{I}_{P_j}$ sufficing that $h \in H : P(h) = P_j$.
6. For each party P_j, a utility function U_j defined on Z denotes his payoffs.

Another important notion in game theory, is equilibria, which guarantee parties not to deviate from specific strategy. The main task of sub-game perfect Nash equilibrium in extensive and incomplete information is to delete unbelievable threat strategies from Nash equilibria such that reasonable results are predicted. More specifically, sub-game perfect Nash equilibrium requires that the behaviors in equilibrium are optimal in each information set.

Definition 2. A strategy profile is a sub-game perfect Nash equilibrium if this profile is Nash equilibrium for each sub-game.

One way to solve equilibria in extensive games under incomplete information is backward induction. Backward induction is an iterative process, which is commonly used in finite extensive form and sequential games. More specifically, the player who makes the last move in the game choose his/her optimal strategy. Then, given these optimal strategy, it's turn for the next-to-last moving player to choose his/her optimal strategy. This process continues backward until each player involved in this game choose their optimal strategies. In effect, the optimal strategy for each player constitutes the strategy profile (aka. sub-game perfect Nash equilibrium).

Theorem 1. A strategy profile is sub-game perfect Nash equilibrium of extensive game with incomplete information, if and only if it's selected by back-

ward induction.

2.2. Policies in mixing coins

Mixing coins is proposed to strengthen anonymity (avoiding privacy leakage), since the transactions and their origins are public and can be easily traced with blockchain exploration tools. The mixing process is as follows [3], taking Coinjoin as an example.

1. Assume a group of parties $U = u_1, u_2, ...u_n$ who are willing to mix their coins. They resort to service providers, e.g. some wallets, which provide mixing service. Coinjoin is one of these services, is practical and implemented in some wallet [24, 25].
2. Policies about mixing with bad coins are in place and publicly known by the parties. That is, the parties should know the consequence if they mix with bad coins. Thus, they decide whether to mix according to the quality of their coin. Generally, parties with bad coins are willing to mix so that they can laundry their coins. On the other hand, parties with good coins risk to lose their money if they mix with bad coins. Therefore, the mixing policies are rather important in the mixing process.
3. With the help of the wallet, all u_i generate a mixing transaction that includes the addresses of each u_i as inputs and the mixing addresses of the mixing service as outputs.
4. The mixing service needs to be trusted by parties in U, which need to sign with their keys the transaction. of the members in U. Otherwise, the transaction is invalid.
5. Parties verify whether the jointly signed transaction sends correct coins to the output addresses. If not, they refuse to mix coins and quit.

After mixing, it is hard to trace coins by simply relying on transactions' input-output relations, since parties in mixing coins merge their inputs and outputs into a single transaction. Parties who seek anonymity need to find available peers to mix with in the blockchain system. Therefore, they are generally ready to pay a mixing fee to incentive others to take part in mixing with them.

Table 1: The update rules of coins' quality for poison policy

q_a	q_b	q'_a	q'_b
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Issues spring up because perpetrators may be willing to mix their low-quality (bad) coins, both earning the mixing fee and laundry their money. The solution to blacklist unsafe coins is effective, since parties will decline to mix their coins with those in the blacklist in order to avoid that bad coins diffuse through mixing. [22] presents three policies for quality propagation: *poison, haircut* and *seniority* policy. In the sequel, we briefly summarise their basic idea.

We inherit their notation: $q_a, q_b \in [0, 1]$ denote the quality of coins of a privacy seeker and a privacy provider, respectively, where $q_a = 0$ denotes that all coins of the privacy seeker are bad, and $q_a = 1$ that all are good. We use q'_a, q'_b to denote the updated coin quality after the mixing and a, b to denote the number of coins of the parties, respectively.

- **Poison policy.** The presence of a single bad coin "poisons" all coins of a party, i.e. $q_a = 0$ if only one coin is blacklisted. Differently, $q_a = 1$ implies that all coins are good. For simplicity, we assume that the number of output coins equals that of input coins. Coin quality is updated according to Table 1. It's easy to see that $q'_a = q'_b = q_a q_b$.

- **Haircut policy.** The privacy seeker and provider are again jointly responsible for the bad coins, as they will have the same coin quality after the mix, defined as $q'_a = q'_b = \frac{aq_a + bq_b}{a+b}$. Obviously, parties with higher coin quality have no incentives to mix (if not the fees), since their quality will decrease.

- **Seniority policy.** The privacy seeker and privacy provider need to know

the exact coin quality q_a and q_b, in order to enter negotiation. Importantly, the most distinctive feature of the seniority policy is that the coin quality does not change through mixing, after negotiation. This policy however requires that the blacklist is updated timely, otherwise parties fails to negotiate beforehand. A detailed example of negotiation can be found in in [22].

3. Games and utility under incomplete information

3.1. Game with deposits under incomplete information

The framework of our mixing games is similar to that of [22] except for the information sets. There are two parties in each game: privacy seeker $\mathcal{S}$ with a coins of quality q_a, and privacy provider $\mathcal{P}$ with b coins of quality q_b, where q_a and q_b denote the ratio of good coins for $\mathcal{S}$ and $\mathcal{P}$ respectively. Here good coins means coins which are not listed in blacklist. The game is as follows.

1. $\mathcal{S}$ decides whether to sponsor a mixing request and the coins who will invest in the mixing.
2. If $\mathcal{S}$ decides not to mix with others, $\mathcal{S}$ does not to sponsor a request. The ratios of good coins remain unchanged. In general, $\mathcal{S}$ invests a for mixing. However, $\mathcal{S}$ must budge for the bribery c once s/he decides to sponsor the request. Put differently, $\mathcal{S}$ only invest $a - c$ if s/he decides not to sponsor a request. Otherwise, s/he should invest a including the bribery.
3. If $\mathcal{S}$ sponsors a request for mixing coins with cost g. The cost g is burned no matter $\mathcal{P}$ accepts it or not. Note that the invest coins are a and b for $\mathcal{S}$ and $\mathcal{P}$ respectively. Then it's turn for $\mathcal{P}$ to decide whether to accept the request.
   (a) If $\mathcal{P}$ does not accept to mix, two parties fail to mix. The ratios of good coins remain unchanged.
   (b) If $\mathcal{P}$ accepts to mix, $\mathcal{P}$ gets a bribery c.
   (c) Finally, $\mathcal{S}$ and $\mathcal{P}$ update ratios of good coins q'_a, q'_b.
4. So far, the game is identical to [22].

The most distinctive feature is that the blacklist may be updated periodically, i.e. the blacklist is updated during party interaction. In this case, parties with bad coins may take advantage from mixing with others. For example, $\mathcal{S}$ requests to mix a good coins, currently not in the blacklist, with $\mathcal{P}$. $\mathcal{P}$ checks the blacklist learning that the a coins are not in blacklist and accepts the request. However, before they de-facto mix their coins, the blacklist is updated including the a coins as bad coins. Consequently, $\mathcal{P}$ suffers from mixing bad coins without any remedial measure. In some sense, the blacklist provides only partial information to $\mathcal{P}$ in this example.

Ideally, $\mathcal{P}$ should be able to cope with the limits of such partial information from the blacklist. We introduce a deposit enforced by smart contracts in the policies described above, seen as a game. The basic idea is simple: $\mathcal{S}$ deposits d_1 before sponsoring the request and $\mathcal{P}$ deposits d_2 before accepting the request. For simplicity, we assume that $d = d_1 = d_2$. Deposits will be refunded to each party if there are no bad coins after their interaction. Note that it's assumed that blacklist updates before refunding happens, so that partial information becomes common knowledge. Furthermore, in our game $\mathcal{S}$ and $\mathcal{P}$ move alternatively under incomplete information, which differs from the simultaneous move in [22]. Due to partial and evolving information, the information set for $\mathcal{S}$ and $\mathcal{P}$ may differ. In the sequel, parties achieve expected utilities $\overline{U}$ with respect to their information set. The extensive game under incomplete information with deposits, dubbed as G_d, is shown in Figure 2, and defined as $< \mathcal{P}, A_{i\in\{\mathcal{S},\mathcal{P}\}}, H, P, f_c, (\mathcal{I}_{P_i})_{i\in\{\mathcal{S},\mathcal{P}\}}, U_{i\in\{\mathcal{S},\mathcal{P}\}} >$ according to Definition 1:

1. A set of parties $\mathcal{P} = \{\mathcal{S}, \mathcal{P}\}$, denoting privacy seeker and privacy provider respectively.
2. Action set $A_{\mathcal{S}} = \{Sponsor, Not\ Sponsor\}$, where the former denotes that $\mathcal{S}$ sponsor a mixing smart contract and the latter denotes the opposite side. Action set $A_{\mathcal{P}} = \{Accept, Not\ Accept\}$, where the former denotes that $\mathcal{P}$ accept mixing and the latter denotes the opposite side.
3. The set of history H denotes the action sequence from the root to the ter-

minal nodes in Figure 2. For example, (*Sponsor*, *Accept*) and (*Sponsor*, *Not Accept*) are histories in G_d. The terminal nodes with respect to terminal histories Z are graphical represented as squares in Figure 2. Nonterminal nodes are presented as hollowed and solid circles. The hollowed circles mean that decision maker is either $\mathcal{S}$ or $\mathcal{P}$. The solid circle means that blacklist leads to partial information with uncertain probabilities p and $1-p$, the meaning of which varies with policies [26]. We will present details in following sections.

4. The function P assigns to each nonterminal history a party so that they take their actions alternatively. For example, P assigns $\mathcal{P}$ to nonterminal histories *Sponsor* and *Not Sponsor*.
5. The dotted line labeled with $\mathcal{S}$ ($\mathcal{P}$) in Figure 2 denotes the information set $\mathcal{I}_{\mathcal{S}}$ ($\mathcal{I}_{\mathcal{P}}$).
6. The terminal histories of G_d are labelled with the pair of utilities of $\mathcal{S}$ and $\mathcal{P}$, respectively. These are discussed in the next section.

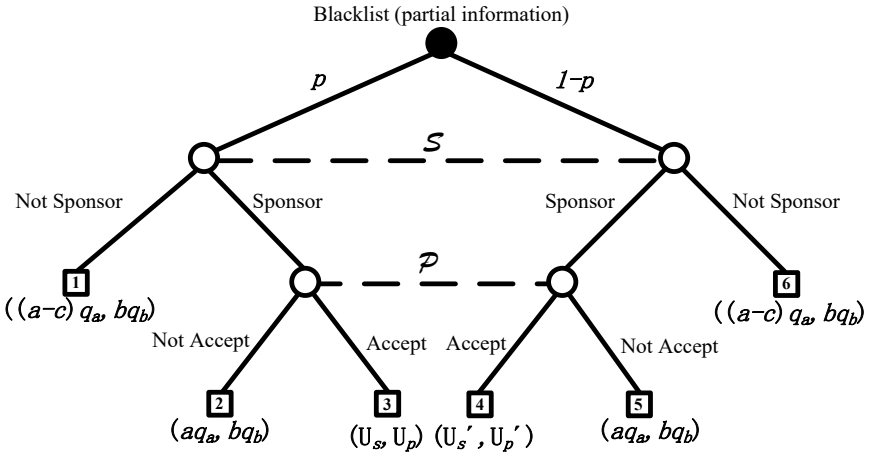

Figure 2: The extensive game under incomplete information with deposits.

3.2. Utilities for the new game

The definition of utilities for each terminal nodes are similar to [22] except for the deposits and laundry incomes. The utility function consists of five components:

1. Privacy incomes u_p: It indicates the privacy achievement for each party with coefficients $\tau_a \in [0,1]$ and $\tau_b \in [0,1]$ for $\mathcal{S}$ and $\mathcal{P}$ respectively, which shows how parties value their privacy. Privacy income contribute to utilities as $\tau_a a q'_a$ and $\tau_b b q'_b$.
2. Laundry incomes u_l: It indicates the degree that parties value the laundering, i.e. mixing, money. The coefficients are $\lambda_a \in [0,1]$ and $\lambda_b \in [0,1]$ for $\mathcal{S}$ and $\mathcal{P}$ respectively. For example, $\mathcal{S}$ does not value the laundering money if $\lambda_a = 0$. Therefore, λ_a only affects the coins after mixing. The utilities contributed by this part are $\lambda_a a q'_a$ and $\lambda_b b q'_b$.
3. Coins for mixing: $\mathcal{S}$ and $\mathcal{P}$ invest a coins and b coins, respectively, to mix with each other. So, the utilities of this part are aq'_a and bq'_b. It is worth remarking that this case includes the mixing costs c (see next item) paid as an incentive to invite $\mathcal{P}$ to mix. However, $\mathcal{P}$ has the right to accept it or not. If $\mathcal{P}$ refuses, then $\mathcal{S}$ only invests $a - c$ coins to mix. In this case, according to the chosen notation, the utilities for $\mathcal{S}$ and $\mathcal{P}$ are $(a-c)q_a$ and bq_b, respectively (as in node 1 in Figure 2).
4. Mixing costs c: This cost is due to $\mathcal{S}$ trying to incentives $\mathcal{P}$ to accept to mix coins together. This is done by sponsoring a smart contract in charge of guaranteeing the promised payment to $\mathcal{P}$. Therefore, the utilities of this part are $-cq_a$ and cq'_b for $\mathcal{S}$ and $\mathcal{P}$ respectively. Note cq'_b instead of cq_b, due to the fact that c coins are paid to $\mathcal{P}$ after mixing. Actually, in the presence of a smart contract, we should also consider the cost g of the gas needed for running the contract. Given that generally g is much smaller than c, we neglected g in this paper for simplicity.
5. Deposits u_d: Deposits are introduced for the purpose of preventing parties from using bad coins in mixing: parties may lose the deposit if their coins

result to be in the blacklist after an update of the list. As an example, $\mathcal{S}$ wants to mix $coin_{\mathcal{S}}$ coins, which are not in the blacklist L at the beginning of the process, but will be included in the revised blacklist L' (Figure 1). $\mathcal{P}$ will accept mixing since $coin_{\mathcal{S}}$ are good coins. Thus, $\mathcal{P}$ will eventually suffer from mixing with coins that have been meanwhile blacklisted. Under complete information, $\mathcal{P}$ would instead have never accepted the mixing. $\mathcal{S}$ may leverage the updating gap to decoy $\mathcal{P}$ for mixing. Therefore, the deposits are important to prevent $\mathcal{S}$ from a successful decoy.

After the mixing, no deposit will be refunded to any part that will result to have used bad coins in the mixing. If only one party will result to have used bad coins, a fee will be deducted from its (non-returned) deposit and paid to the counterpart, which does not have bad coins, as a compensation, i.e. a *bonus*. The amount of the bonus is proportional to the amount of bad coins used in the mixing by the (only) offender. If both parties will result to have used bad coins, the fee will be deducted to both - proportionally - and paid as a bonus to the counterpart, respectively.

For instance, suppose that in the end, according to the revised blacklist L', we have $q'_a = 0.333$, i.e. only 1/3 of $\mathcal{S}$'s coins are good, while all $\mathcal{P}$'s coins are good. Then 2/3 of $\mathcal{S}$'s deposit, 6 out of 9 coins say, will be paid to $\mathcal{P}$ as a bonus.

It is worth remarking that *i.* deposits and bonus are fully managed by a smart contract according to the updated blacklist L' (Figure 1). Parties pay the deposit to the smart contract, since they can trust that it will be correctly executed by the blockchain. Different formulations for refunds and bonuses can be adopted; *ii.* the contract earns coins from the non-returned deposits. One could imagine that such a surplus could be used for supporting parties that have incurred losses due to bad mixing. This is scope for future work; and *iii.* for the sake of simplicity we assume here that deposit coins are actually good coins (also in the revised L'). Relaxing this assumption is also scope for future work.

Formally, the depositing and refunding processes are as follows.

(a) Parties do not pay a deposit if $\mathcal{S}$ does not sponsor or $\mathcal{P}$ does not accept the mixing (nodes 1,2,5,6 in Figure 2).

(b) If $\mathcal{S}$ sponsors a smart contract and $\mathcal{P}$ accepts it, then $\mathcal{S}$ and $\mathcal{P}$ pay the deposit d to the smart contract *before* mixing. Therefore, deposits incur $-dq_a$ and $-dq_b$ to $\mathcal{S}$ and $\mathcal{P}$ respectively.

(c) The utility due to deposits depends on whether parties get refunds or a bonus after mixing, according to the following three possible cases:

i. Both $\mathcal{S}$ and $\mathcal{P}$ have no bad coins after mixing. The smart contract refund d to both parties. Therefore, deposits incur $-dq'_a$ and $-dq'_b$ to $\mathcal{S}$ and $\mathcal{P}$ respectively. So, the net utility incurred by depositing are $-dq_a + dq'_a$ and $-dq_b + dq'_b$, respectively. Note that $-(-dq_a + dq'_a) = dq_a - dq'_a$ and $-(-dq_b + dq'_b) = dq_b - dq'_b$ are withhold by the smart contract as bonus to others.

ii. Only one party, $\mathcal{S}$ say, has bad coins. The smart contract returns to $\mathcal{P}$ its deposit d and a bonus deducted from the d of $\mathcal{S}$. It returns the remaining coins from $\mathcal{S}$'s deposit to $\mathcal{S}$. The bonus is computed according to coin qualities: $\mathcal{P}$ receives dq'_b (due to refund of the deposit) and $dq_a - dq'_a$ (due to a positive bonus). $\mathcal{S}$ does not get the deposit back. Therefore, the net utility for $\mathcal{S}$ is $-dq_a$ and for $\mathcal{P}$ is $(-dq_b + dq'_b) + (dq_a - dq'_a)$.

iii. Both $\mathcal{S}$ and $\mathcal{P}$ have bad coins. In this case, the smart contract does not refund any deposit and the bonuses are determined as follow: the bonus for $\mathcal{S}$ is $dq_b - dq'_b$ and for $\mathcal{P}$ is $dq_a - dq'_a$. Therefore, the net utility for $\mathcal{S}$ is $(-dq_a + dq'_a) + (dq_b - dq'_b)$ and for $\mathcal{P}$ is $(-dq_b + dq'_b) + (dq_a - dq'_a)$.

In Figure 2, the utilities for $\mathcal{S}$ and $\mathcal{P}$ are listed below the leaf nodes. The utilities for nodes 3 and 4 are more complex, and depend on the different policies. Note that the utilities for each leaf do not necessarily consist of all the five parts mentioned above.

- Utilities for nodes 1 and 6. $\mathcal{S}$ does not sponsor a mixing request. Therefore, $\mathcal{S}$ only spends $a - c$ coins, not needing the sponsoring budget c. On

the other hand, $\mathcal{P}$ retains b coins. The coins' quality remains unchanged. Therefore, the utilities for node 1 and 6 are defined as $(a-c)q_a$ and bq_b, respectively.

- Utilities for nodes 2 and 5. $\mathcal{S}$ sponsors a mixing request while $\mathcal{P}$ does not accept the request. It follows that $\mathcal{S}$ and $\mathcal{P}$ do not mix. However, $\mathcal{S}$ should budget for sponsoring beforehand, even if $\mathcal{P}$ does not accept the request. $\mathcal{P}$ retains b coins. In addition, the coins' quality remains unchanged. So the utilities for node 2 and 5 are defined as aq_a and bq_b, respectively.

- Utilities for node 3 and 4. $\mathcal{S}$ sponsors a smart contract and $\mathcal{P}$ accepts it, and $\mathcal{S}$ and $\mathcal{P}$ mix coins. Recall that $\mathcal{S}$ bribes $\mathcal{P}$ with c. Thus, $\mathcal{S}$ and $\mathcal{P}$ invest a (which includes c) and b, respectively. Furthermore, coins' qualities are updated according to the different policies. Distinguishing in our framework, parties deposit a deposit, whose restitution is regulated by the smart contract according to the updated blacklist L', so coping with partial information and uncertainty. Utilities in nodes 3 and 4 differ according to the coins' quality, but they can be expressed by the general formulas $\hat{U}_{\mathcal{S}}$ and $\hat{U}_{\mathcal{P}}$. In the sequel, $U_{\mathcal{S}}$, $U_{\mathcal{P}}$, $U'_{\mathcal{S}}$ and $U'_{\mathcal{P}}$ are instances of (1) and (2), with different q_a, q_b, q'_a, and q'_b. Let $\beta_a = \tau_a + \lambda_a + 1$, $\beta_b = \tau_b + \lambda_b + 1$.

$$\begin{aligned}\hat{U}_{\mathcal{S}} &= \tau_a a q'_a + \lambda_a a q'_a + (a q'_a - c q_a) + (-d q_a + d q'_a) + (d q_b - d q'_b) \\ &= \beta_a a q'_a - c q_a - d q_a + d q'_a + d q_b - d q'_b\end{aligned} \tag{1}$$

$$\begin{aligned}\hat{U}_{\mathcal{P}} &= \tau_b b q'_b + \lambda_b b q'_b + (b q'_b + c q'_b) + (-d q_b + d q'_b) + (d q_a - d q'_a) \\ &= \beta_b b q'_b + c q'_b - d q_b + d q'_b + d q_a - d q'_a\end{aligned} \tag{2}$$

4. Equilibria in mixing games under incomplete information

In this section, we will analyze the conditions for sub-game perfect Nash equilibrium with respect to each policy under incomplete information by implementing backward induction. We will present, given proper conditions,

(*Sponsor*, *Accept*) is sub-game perfect Nash equilibrium for each policy according to backward induction.

4.1. Poison policy

According to this policy, $q_a = 1$ holds when none of the a coins is in the blacklist. q_a, q_b, q'_a and q'_b are defined in Table 1. Furthermore, it holds that $q'_a = q'_b = q_a q_b$.

However, $\mathcal{P}$ may not be sure whether q_a could actually evolve to $q'_a = 0$ after a revision of the blacklist, given partial information of $\mathcal{P}$. $\mathcal{P}$ has a prior probability p on whether $q_a = 1$. Note that, in node 3, q_a is assumed to be 1 with probability p. Therefore, the utilities for $\mathcal{S}$ and $\mathcal{P}$ follow from (1) and (2) with $q_a = 1$ and $q'_a = q'_b = q_a q_b = q_b$:

$$U_{\mathcal{S}} = \beta_a a q_b - c - d + d q_b, \quad U_{\mathcal{P}} = \beta_b b q_b + c q_b \tag{3}$$

Note that formula (3) may correspond to 5(d) case 1 or 5(d) case 2 relying on coins' quality of $\mathcal{P}$.

In node 4, q_a is assumed to be 0 with probability $1-p$. Therefore, the utilities for $\mathcal{S}$ and $\mathcal{P}$ follow from (1) and (2) with $q_a = 0$ and $q'_a = q'_b = q_a q_b = 0$:

$$U'_{\mathcal{S}} = U'_{\mathcal{P}} = 0 \tag{4}$$

Note that formula (4) may correspond to 5(d) case 2 or 5(d) case 3 relying on coins' quality of $\mathcal{P}$.

It is worth remarking that $\mathcal{P}$ may accept mixing sponsored by $\mathcal{S}$ only when $q_b = 1$. Otherwise, $U_{\mathcal{S}} = -c - d < 0 < b q_b$ and $\mathcal{S}$ would not propose a mixing to $\mathcal{P}$. Therefore, we have,

$$U_{\mathcal{S}} = \beta_a a - c, \quad U_{\mathcal{P}} = \beta_b b q_b + c q_b \tag{5}$$

Consequently, we first analyse the choices of $\mathcal{P}$ according to backward induction. $\mathcal{P}$ had to make a decision once G_b reaches information set $I_{\mathcal{P}}$. Here $\mathcal{P}$ may either accept or not accept to mix coins with $\mathcal{S}$.

1. If he chooses *Not Accept*, $\mathcal{P}$ gets expected utility bq_b no matter what q is. Note that $\mathcal{P}$ gets expected utility due to the partial information. For example, $\mathcal{P}$ think $\mathcal{S}$ has bad coins with probability p. Therefore, the utility is bq_b in node 2 with probability p and bq_b in node 5 with probability $1-p$ (ref. Figure. 2). The expected utility is $pbq_b + (1-p)bq_b = bq_b$.
2. If he chooses *Accept*, he gets expected utility $\overline{U}_{\mathcal{P}} = pU_{\mathcal{P}} + (1-p)U'_{\mathcal{P}}$. Note that in node 3, the utility is $U_{\mathcal{P}}$ with probability p and in node 4, the utility $U'_{\mathcal{P}}$ with probability $1-p$.

The expected utilities are defined in similar way if not specified. $\mathcal{P}$ chooses *Accept* if $\overline{U}_{\mathcal{P}} > bq_b$. That is,

$$p > \frac{b}{b\beta_b + c} \quad (when \;\; q_b = 1) \tag{6}$$

Note that $\mathcal{P}$ can choose any action when $q_b = 0$ since both lead to utility 0.

Then we backward to anaylse $\mathcal{S}'s$ choices.

1. If he chooses *Not Sponsor*, $\mathcal{S}$ gets utility $a - c$.
2. If he chooses *Sponsor*, he gets expected utility $\overline{U}_{\mathcal{S}} = pU_{\mathcal{S}} + (1-p)U'_{\mathcal{S}}$.

$\mathcal{S}$ chooses *Sponsor* if $\overline{U}_{\mathcal{S}} > a - c$. That is,

$$p > \frac{a-c}{\beta_a - c} \quad (when \;\; q_b = 1) \tag{7}$$

Given $p > \frac{b}{b\beta_b + c}$, $p > \frac{a-c}{\beta_a - c}$, $(Sponsor, Accept)$ is sub-game perfect Nash equilibrium. That is, no one can increase his/her utility by unilaterally deviate from the equilibrium.

4.2. Haircut policy

In haircut policy, we have $q'_a = q'_b = \frac{aq_a + ba_b}{a+b}$.

1. $\mathcal{S}$ obtains expected utility $\overline{U}_{\mathcal{S}} = \beta_a a \frac{aq_a + bq_b}{a+b} + (q_a - q_b)d$ if he choose *Sponsor* and aq_a otherwise.

2. G_b will follow the history $(Sponsor, Accept)$, then it should suffice that $\overline{U}_{\mathcal{P}} > bq_b$ and $\overline{U}_{\mathcal{S}} > aq_a$. That is,
455 $\mathcal{P}$ obtains expected utility $\overline{U}_{\mathcal{P}} = \beta_b b\frac{aq_a+bq_b}{a+b} + (q_a - q_b)d$ if he choose *Accept* and bq_b otherwise.

G_b will follow the history $(Sponsor, Accept)$, then it should suffice that $\overline{U}_{\mathcal{P}} > bq_b$ and $\overline{U}_{\mathcal{S}} > aq_a$. That is,

$$\frac{q_a}{q_b} < \frac{ab\beta_a - (a+b)d}{(1-\beta_a)a^2 - (a+b)d + ab} \tag{8}$$

$$\frac{q_a}{q_b} > \frac{(1-\beta_b)b^2 - (a+b)d + ab}{ab\beta_b + (a+b)d} \tag{9}$$

If $\frac{q_a}{q_b}$ meets formulas (8) and (9), $\mathcal{S}$ has incentives to sponsor a mixing request
460 and $\mathcal{P}$ has incentives to accept it. There also exists a sub-game equilibrium $(Sponsor, Accept)$ in haircut policy.

4.3. Suicide policy: a variant of seniority policy

Seniority policy is invalid in G_d because blacklist may be partial information and $\mathcal{S}$ and $\mathcal{P}$ fail to negotiate with respect to allocation without exactly knowing
465 q_a and q_b. A trivial solution is to leverage cryptographic tools (e.g. secure multiparty computation) for parties to negotiate. For example, parties securely compute a function $f(q_a, q_b) = (q'_a, q'_b)$ with their private inputs. However, $\mathcal{S}$ and $\mathcal{P}$ may conceal the coin quality. Furthermore, efficiency may be not acceptable for specific applications.

470 In this paper, we propose a variant of seniority policy, which allows parties to mix coins without negotiating beforehand. The basic idea is simple: the party who has higher ratio of bad coins will pay the bill of all bad coins. For example, if $\mathcal{S}$ has more bad coins, all bad coins will be assigned to $\mathcal{S}$ after mixing. The update rules of coins' quality for suicide policy is shown in Table 2.

The utilities for each terminal history are defined similar to poison policy. The utility definitions are shown in Formulations (10)-(13).

$$U_{\mathcal{S}} = (\beta_a a + d + \frac{ad}{d}) - (c + d + \frac{ad}{d})q_a \tag{10}$$

Table 2: The update rules of coins' quality for suicide policy

	q'_a	q'_b
$q_a \geq q_b$	1	$max\{0, \frac{bq_b+a(q_a-1)}{b}\}$
$q_a < q_b$	$max\{0, \frac{aq_a+b(q_b-1)}{a}\}$	1

$$U_{\mathcal{P}} = (\beta_b b + c + d)(q_b + \frac{a}{b}(q_a - 1)) - q_b d + q_a d - d \tag{11}$$

$$U'_{\mathcal{S}} = (\beta_a a - c + d)(q_a + \frac{b}{a}(q_b - 1)) - q_a d + q_b d - d \tag{12}$$

$$U'_{\mathcal{P}} = (\beta_b b + c + d) - q_b d + \frac{bd}{a}(1 - q_b) \tag{13}$$

Formulations (10)-(13) are too complex to analyse, so we present a simple version for suicide policy. More specifically, we assume $\frac{bq_b+a(q_a-1)}{b} < 0$ and $\frac{aq_a+b(q_b-1)}{b} < 0$ are always less than 0. That is, $aq_a + bq_b < min\{a, b\}$. Then $max\{0, \frac{bq_b+a(q_a-1)}{b}\} = 0$ and $max\{0, \frac{aq_a+b(q_b-1)}{b}\} = 0$. The simplified suicide policy is shown in Table 3. Consequently, we have the simplified utilities.

$$U_{\mathcal{S}} = \beta_a a - cq_a + d - q_a d + q_b d > 0 \tag{14}$$

$$U_{\mathcal{P}} = -q_b d + q_a d - d < 0 \tag{15}$$

$$U'_{\mathcal{S}} = -cq_a - q_a d + q_b d - d < 0 \tag{16}$$

$$U'_{\mathcal{P}} = \beta_b b + c + d - q_b d + q_a d > 0 \tag{17}$$

Backward induction. $\mathcal{P}$ may either accept or not accept to mix coins with $\mathcal{S}$.

Table 3: The update rules of coins' quality for suicide policy

	q'_a	q'_b
$q_a \geq q_b$	1	0
$q_a < q_b$	0	1

1. If he chooses $Not\ Accept$, $\mathcal{P}$ gets expected utility bq_b no matter what q is. The utilities are bq_b in nodes 2, 5 (ref. Figure 2). Therefore, the expected utility is $pbq_b + (1-p)bq_b = bq_b$.
2. If he chooses $Accept$, he gets expected utility $\overline{U}_{\mathcal{P}} = pU_{\mathcal{P}} + (1-p)U'_{\mathcal{P}}$. Note that the utility is $U_{\mathcal{P}}$ in nodes 3 and $U'_{\mathcal{P}}$ in node 4.

$\mathcal{P}$ chooses $Accept$ if $\overline{U}_{\mathcal{P}} > bq_b$ suffices. That is,

$$q_a > \frac{bq_b + dq_b + bp\beta_b + pc + 2pd - b\beta_b - c - d}{d} \tag{18}$$

Then we backward to analyse $\mathcal{S}'s$ choices.

1. If he chooses $Not\ Sponsor$, $\mathcal{S}$ gets utility aq_a.
2. If he chooses $Sponsor$, he gets expected utility $\overline{U}_{\mathcal{S}} = pU_{\mathcal{S}} + (1-p)U'_{\mathcal{S}}$.

$\mathcal{S}$ chooses $Sponsor$ if $\overline{U}_{\mathcal{S}} > aq_a$ suffices. That is,

$$q_a < \frac{ap\beta_a + (2p + q_b - 1)d}{a + c + d} \tag{19}$$

Similarly to poison and haircut policies, the sub-game perfect Nash equilibrium $(Sponsor, Accept)$ exists in suicide policy when formulas (18)(19) are established. Put differently, $\mathcal{S}$ sponsors a mixing request and $\mathcal{P}$ accepts it. Meanwhile, both parties have no incentives to mix with bad coins. Otherwise, the one who mix more bad coins will suffer from affording all bad coins.

5. Simulations

We simulate the conditions of aforementioned equilibria for each policy with Matlab. In this paper, we address extensive games under incomplete information, a more complex but practical scenario. For simplicity, we fix some parameters, only keeping crux metrics in order to find the incidence of equilibria.

Furthermore, we restrict the range for the parameters without loss of generality to simulate the theoretical results. For example, $a, b \in [1, 10]$, $q_a, q_b, p \in [0, 1]$, $\beta_a, \beta_b \in [1, 3]$, $d > a$, $d > b$, $1 < c < a$. We highlight that in [22], there is no mixing equilibrium for these policies. While in this paper, there are, given proper parameters, no more than one equilibrium for all three policies.

5.1. Poison policy

Recall that in simultaneous-move game under imperfect information, the authors claim that no one would like to mix good coins with others in case of encountering bad coins. That is, no equilibrium exist in [22], where privacy seeker sponsors mixing and privacy provider responds to it. However, in this paper, we prove that there are more than one equilibrium in G_d for the case of poison policy. More specifically, the Formulas (6) and (7) are borderlines for $\mathcal{P}$ and $\mathcal{S}$ respectively to reach the equilibrium. A more clear graphical representation is shown in Figure 3. For example, $c = 7.25$, $p = 0.35$ is an equilibrium here. In effect, the points in gray area constitute the set of equilibria. The reason for the absence of q_a and q_b is that their values are either 0 or 1.

The trend for the grey area is: c increases with the decreasing of p. That is, $\mathcal{S}$ costs less if $\mathcal{P}$ has more confidence that $\mathcal{S}$ has more good coins. An extreme case in Figure 3 occurs that $\mathcal{P}$ would like to accept the mixing request when he believes, with probability only around 60%, that $\mathcal{S}$ has good coins even if the payment is 0. However, the payment increases to 7.25 when the probability decreases to 35%. We may take another way to understand: a bad reputation for $\mathcal{S}$ should pay a high cost for mixing. On the other hand, $\mathcal{S}$ has no incentives to sponsor a mixing when cost is higher than 7.25.

Note that $p = 0.5$ is chosen since not all choices of p lead to the existence of equilibria. Figure 4 presents the relationship between p and q_a under fixed parameters. q_a is valid only p falls between 0.44 and 0.57, which may variant with the changes of other parameters, while the trend is similar.

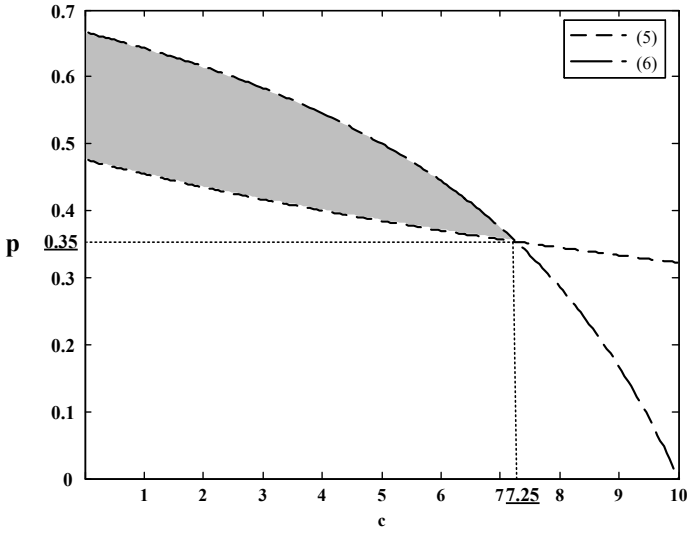

Figure 3: The conditions of equilibrium for poison policy ($a = b = 10$, $d = 25$, $\beta_a = 1.5$, $\beta_b = 2.1$).

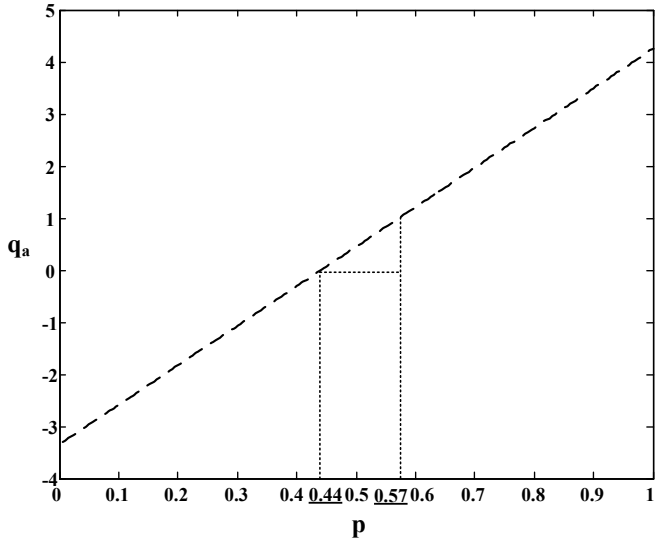

Figure 4: The relationship between p and q_a ($q_b = 0.5$, $a = 10$, $d = 25$, $\beta_a = 1.5$, $\beta_b = 2.1$, $c = 5$.

5.2. Haircut policy

Recall that in simultaneous-move game under imperfect information with respect to haircut policy, the authors claim that "there will be no equilibrium outcome with a strictly positive payoff for both players, ..."[22]. In this paper, Figure 5 illustrates the effect of a, q_a/q_b on equilibria. Unlike the case of poison policy, there is no distinct linear relationship between a and q_a/q_b. The common universality for the points in grey area in Figure 5 is that most points require $q_a/q_b > 1$. That is, the quality of $\mathcal{S}$ should be at least higher than that of $\mathcal{P}$. For instance, there exists an equilibrium when $a = 9$ and $q_a/q_b = 1.22$. Note that it's not always establish for haircut policy, where, for example, $a = 2, 3, 4...$ and $q_a/q_b = 1.22$. Therefore, the existence of equilibria heavily depends on parameters.

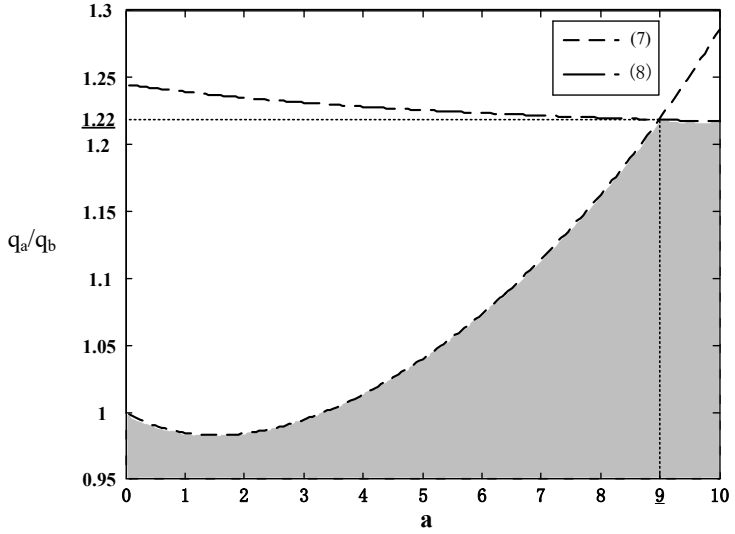

Figure 5: The conditions of equilibrium for haircut policy ($b = 10$, $d = 20$, $\beta_a = 1.5$, $\beta_b = 0.5$, $c = 0.1$, $p = 0.5$).

5.3. Suicide policy

In [22], the seniority policy can be reduced to haircut policy. Therefore, it inherits the conclusions of haircut policy there. The authors also claim that

seniority policy is more or less a modification of signal game under incomplete information with cost commitment. Unfortunately, there are still no more mixing equilibria for seniority with respect to seniority policy. In this paper, we revisit seniority policy and propose an updated version since parties cannot negotiate without knowing the exact values of bad coins. The deposits in suicide policy can be regarded as cost commitment for game G_d. The existence of deposits deters parties not to mix bad coins. Otherwise deposits will not refund. There are more than one equilibrium for suicide policy under the thrust of deposits (grey area in Figure 6). For example, $c = 2.8$, $q_a = 0.56$ constitute an equilibrium. Generally, c is relatively low when q_a is high. That is, $\mathcal{S}$ would like to pay less if he has more good coins. On the other hand, he must pay more for his bad coins. Otherwise, $\mathcal{P}$ would not take risk to accept mixing bad coins. No equilibrium exists when q_a is lower than 0.5 even if $\mathcal{S}$ pays all his coins. So the existence of equilibria heavily depends on the parameters.

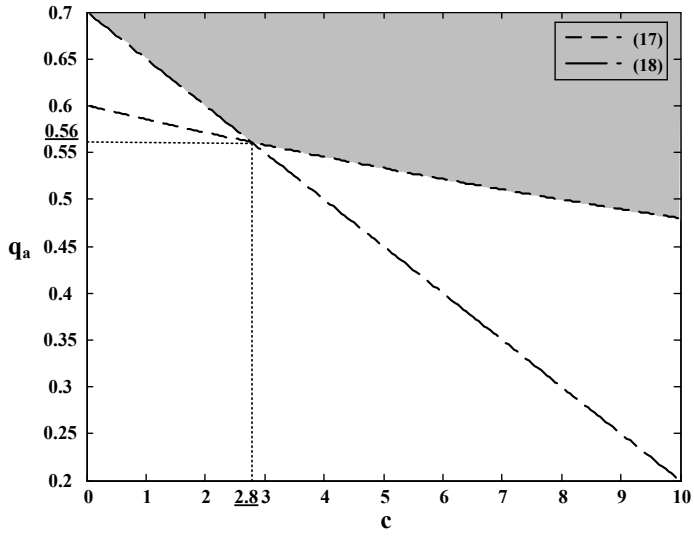

Figure 6: The conditions of equilibrium for suicide policy ($a = 5$, $b = 10$, $d = 25$, $\beta_a = 1.5$, $\beta_b = 2.1$, $q_b = p = 0.5$).

We fix $q_b = 0.6$ as a reference for the case $q_b = 0.5$, where other parameters

are the same (as shown in Figure 7). The equilibria for $q_b = 0.6$ include the points in area IV, while those for $q_b = 0.5$ include the points in areas I, II, III and IV. It's obvious the equilibria area is larger when q_b is smaller, which means $\mathcal{P}$ prefer to mix when he has more bad coins. Note that here we only discuss the existence of equilibria instead of precise values and the equilibria areas vary with variant parameters.

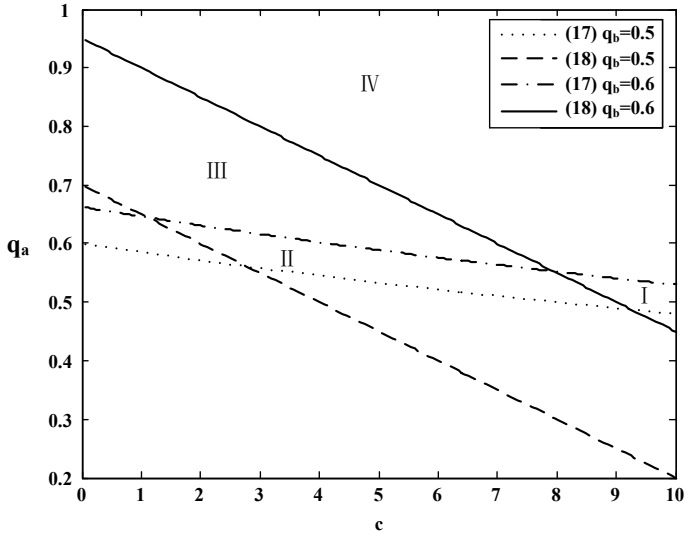

Figure 7: The conditions of equilibrium for suicide policy ($a = 5$, $b = 10$, $d = 25$, $\beta_a = 1.5$, $\beta_b = 2.1$, $q_b = p = 0.5$).

6. Conclusions and future works

The method of mixing coins is a double-edged sword, which may provide privacy and facilitate money launder. Previous works solve this problem when blacklist is common knowledge or present simultaneous game under imperfect information. In this paper, we consider complex scenario, where blacklist is not updated timely. Furthermore, we allow parties to take actions alternatively. Therefore, previous solutions do not match to this new scenario. We introduce deposits enforced by smart contracts and extend the utilities in the new model.

Furthermore, we analyze poison and haircut policies under incomplete information and list the conditions for sub-game perfect equilibrium. That is, $\mathcal{S}$ sponsors a smart contract for mixing coins and $\mathcal{P}$ accepts it. However, deposits mechanism fails to directly transplant for seniority policy since parties cannot negotiate beforehand. In the sequel, we proposed suicide policy to prevent bad coins from entering mixing.

In this paper, we fixed our model under some assumptions. For example, we assume that the blacklist will be updated after the game. These assumptions should be unbundled and more general models are needed in the future works. Furthermore, the parameters in our model are not necessary optimal ones since we choose them manually. In the future works, we may break this defect by introducing machine learning. One of the urgent problems is to build a data set for the target.

Acknowledgments

This study was funded by Foundation of National Natural Science Foundation of China (grant number:61771231, 6150028), Natural Science Shandong Province (grant number: ZR2016FM23, ZR2017MF010, ZR2017MF062), Key Research and Development Program of Shandong Province NO. 2019GGX101025).

References

[1] G. Maxwell, Coinjoin: Bitcoin privacy for the real world. post on bitcoin forum, accessed on August 2013. https://bitcointalk.org/index.php?topic=279249 (2013).

[2] Wiki, Coinjoin, accessed on June 2020. https://en.bitcoin.it/wiki/CoinJoin (2020).

[3] T. Ruffing, P. Moreno-Sanchez, A. Kate, Coinshuffle: Practical decentralized coin mixing for bitcoin.

[4] Y. Wang, A. Bracciali, T. Li, F. Li, X. Cui, M. Zhao, Randomness invalidates criminal smart contracts, Information Sciences 447 (2019) 291–301.

[5] L. Zhang, Y. Wang, F. Li, Y. Hu, M. H. Au, A game-theoretic method based on q-learning to invalidate criminal smart contracts, Information Science 498 (2019) 144–153.

[6] Z. Yang, T. Ouyang, X. Fu, X. Peng, A decision making algorithm for online shopping using deep learningcbased opinion pairs mining and q-rung orthopair fuzzy interaction heronian mean operators, International Journal of Intelligent Systems 35 (5).

[7] X. Yu, H. Wang, Zheng, X. Zheng, Y. Wang, Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments, International Journal of Ad-Hoc Ubiquitious Computing 23 (3-4) (2016) 14437–151.

[8] Y. Wang, M. Zhao, Y. Hu, Y. Gao, X. Cui, Secure computation protocols under asymmetric scenarios in enterprise information system, Enterprise Information Systems (2019) 1–21.

[9] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, Y.-a. Tan, Secure multi-party computation: theory, practice and applications, Information Sciences 476 (2019) 357–372.

[10] R. Urena, G. Kou, J. Wu, F. Chiclana, E. Herrera-Viedma, Dealing with incomplete information in linguistic group decision making by means of interval type-2 fuzzy sets, International Journal of Intelligent Systems 34 (6) (2019) 1261–1280.

[11] X. Ding, H. Liu, A new approach for emergency decision-making based on zero-sum game with pythagorean fuzzy uncertain linguistic variables, International Journal of Intelligent Systems 34 (7) (2019) 1667–1684.

[12] N. Tsalis, N. Virvilis, A. Mylonas, T. Apostolopoulos, D. Gritzalis, Browser blacklists: the utopia of phishing protection, in: International Conference on E-Business and Telecommunications, Springer, 2014, pp. 278–293.

[13] M. Vasek, M. Weeden, T. Moore, Measuring the impact of sharing abuse data with web hosting providers, in: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, ACM, 2016, pp. 71–80.

[14] B. Edwards, T. Moore, G. Stelle, S. Hofmeyr, S. Forrest, Beyond the blacklist: modeling malware spread and the effect of interventions, in: Proceedings of the 2012 New Security Paradigms Workshop, ACM, 2012, pp. 53–66.

[15] S. Hofmeyr, T. Moore, S. Forrest, B. Edwards, G. Stelle, Modeling internet-scale policies for cleaning up malware, in: Economics of Information Security and Privacy III, Springer, 2013, pp. 149–170.

[16] A. Acquisti, R. Dingledine, P. Syverson, On the economics of anonymity, in: International Conference on Financial Cryptography, Springer, 2003, pp. 84–102.

[17] M. Möser, R. Böhme, D. Breuker, Towards risk scoring of bitcoin transactions (2014) 16–32.

[18] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, E. W. Felten, Mixcoin: Anonymity for bitcoin with accountable mixes, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 486–504.

[19] H. Garg, R. Arora, Generalized intuitionistic fuzzy soft power aggregation operator based on t-norm and their application in multicriteria decision-making, International Journal of Intelligent Systems 34 (2) (2019) 215–246.

[20] W. Yang, Y. Pang, New q-rung orthopair fuzzy partitioned bonferroni mean operators and their application in multiple attribute decision making, International Journal of Intelligent Systems 34 (3) (2019) 439–476.

[21] X. Yu, W. Feng, H. Wang, Q. Chu, Q. Chen, An attention mechanism and multi-granularity-based bi-lstm model for chinese q&a system, Soft Computing 24 (2020) 5831C5845.

[22] R. B. Svetlana Abramova, Pascal Schottle, Mixing coins of different quality: A game-theoretic approach, in: Financial Cryptography Workshops 2017, Springer, 2017, pp. 280–297.

[23] M. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, Cambridge, 2004.

[24] W. van der Laan, Implement coinjoin in wallet. github issue 3226 of official bitcoin repository, https://github.com/bitcoin/bitcoin/issues/3226 (2020).

[25] M. J. W. C. H. P. T. A. e. a. Buterin, V., Dark wallet, https://darkwallet.unsystem.net (2020).

[26] A. Srivastava, L. Kaur, Uncertainty and negation-information theoretic applications, International Journal of Intelligent Systems 34 (6) (2019) 1248–1260.