Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator

Aditya Asgaonkar
Viterbi School of Engineering[*]
University of Southern California
Los Angeles, USA
f20150043@goa.bits-pilani.ac.in

Bhaskar Krishnamachari
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
bkrishna@usc.edu

***Abstract*—A fundamental problem for electronic commerce is the buying and selling of digital goods between individuals that may not know or trust each other. Traditionally, this problem has been addressed by the use of trusted third-parties such as credit-card companies, mediated escrows, legal adjudication, or reputation systems. Despite the rise of blockchain protocols as a way to send payments without trusted third parties, the important problem of exchanging a digital good for payment without trusted third parties has been paid much less attention. We refer to this problem as the Buyer and Seller's Dilemma and present for it a dual-deposit escrow trade protocol which uses double-sided payment deposits in conjunction with simple cryptographic primitives, and that can be implemented using a blockchain-based smart contract. We analyze our protocol as an extensive-form game and prove that the Sub-game Perfect Nash Equilibrium for this game is for both the buyer and seller to cooperate and behave honestly. We address this problem under the assumption that the digital good being traded is known and verifiable, with a fixed price known to both parties.**

***Index Terms*—Blockchain-based Applications, Escrow, Smart Contract, Trustless Payment**

I. Introduction

A fundamental problem for electronic commerce has been the exchange of a digital good for payment. The earliest solutions for this problem date back at least to the earliest days of the world wide web [1], online stores accepting credit card payments for downloadable content. Because the exchange of the good and payment cannot happen simultaneously, there is an inherent tension and need for trust in the trade — the seller must trust that the buyer will pay and the buyer must trust that the seller must deliver. Traditionally, this need for trust has been addressed by introducing a trusted third party — this could be a credit card company, a third party mediator for an escrow [2], legal adjudication or arbitration of disputes [3] or the use of a reputation system to build trust by allowing parties to gain some understanding of the prior behavior of the other [4]. Indeed underground economies where such trust is difficult to obtain are often rife with scams [5].

The blockchain revolution that was ushered in by the publication of the technical paper on Bitcoin [6] has allowed for the first time for digital payments to be made between parties without requiring a trusted third party. However, the problem of exchanging a digital good for payment without requiring a trusted third party has not been widely addressed. In principle, agreements between parties can be codified in the form of smart contracts; however, even recent work on applying smart contract to such transactions has continued to rely on third party mediation. For example, Goldfelder *et al.* [2] consider the same problem and propose the use of smart contracts that involves a third-party mediator or a group of third party mediators, while proving certain security and privacy enhancements over traditional approaches.

In contrast to that prior work, we propose here a smart contract that is deployed by the seller. This smart contract requires both the seller and the buyer to place a sufficiently high deposit into the smart contract. In our proposed protocol, seller first submits its deposit, then the buyer sends payment as well as its own deposit; the seller then sends a key to unlock the digital good. The buyer verifies the good is received and if all is well, sends an approval message to the smart contract. The deposits made by both parties are returned to them only after a successful trade is completed (the seller sends the correct good and the buyer verifies and approves it). In all other cases at least one of the parties will lose its deposit. The protocol involves no third parties at all. We analyze this dual-deposit escrow trade protocol as an extensive form game and show that honest behavior by both parties is the only sub-game perfect Nash equilibrium.

While the use of one-way security deposits to provide trust for one party with respect to the other is quite common and dates back a long time, particularly in the context of home rentals [7], dual-deposits such as the scheme proposed in

[*]Visiting from BITS-Pilani, Goa.

978-1-7281-1328-9/19/$31.00 ©2019 IEEE.

our protocol are not common [1]. However, after we wrote our first draft of this paper, we became aware of a double-deposit escrow mechanism provided by a decentralized online marketplace called BitBay [8], which we further learned is essentially the same as another system called BitHalo [9]. Bigi *et al.* [10] have formalized Bithalo into a particular double-deposit scheme they refer to as DCSP (for decentralized smart-contract protocol) and analyze it game theoretically. There are some differences from our scheme in this paper: in the BitHalo/BitBay double-deposit escrow scheme buyers and sellers both make a deposit through the client or smart-contract, then exchange the good and payment off-chain, and both deposits are released only if both parties confirm that the transaction was successful, with no other form of dispute resolution. Their scheme does not allow for the delivered digital good to be independently verified. In contrast, in our scheme the smart contract is capable of autonomous verification based on a known hash of the digital good - this allows our proposed smart contract to selectively return the buyer's deposit if a complaint by a buyer is valid and to selectively return the seller's deposit if a complaint by a buyer is found to be invalid.

A closely related but fundamentally different problem that has been addressed previously is that of *Atomic Swaps* [11] used in blockchain systems to exchange on-chain assets or tokens between users using smart-contracts. This can be accomplished in a decentralized manner more easily because the movement of such assets corresponds merely to a change of state (such as changing the owner of assets) on the blockchain. In contrast, sending a digital good to the buyer in an encrypted manner in exchange for payment so that only the buyer can decrypt it does not represent merely a change of state on the blockchain.

A. The Buyer and Seller's Dilemma

Consider a scenario where a Seller is attempting to make a sale of a Product to the Buyer. Two transactions are bound to happen:

- **Delivery of Product**: The Seller delivers the Product to the Buyer.
- **Payment for Product**: The Buyer makes a payment to the Seller.

In any trading platform, one of these two transactions must occur first. Depending on which transaction occurs first, one of these factors of trust is introduced between the Buyer and Seller:

- **Trust of Delivery**: Trust in the Seller that if payment is made first, the Product will be delivered.
- **Trust of Payment**: Trust in the Buyer that if the Product is delivered first, then payment will be made.

In nearly all existing systems, the latter transaction is not guaranteed to occur, but is incentivized through a separate entity — an escrow account, a legal document, or a reputation system. Each of these systems have a third-party trusted actor:

- **Escrow Account**: A third-party trusted mediator is introduced who will hold the payment from the Buyer until the Seller delivers the Product.
- **Legal Document**: A legally binding document is introduced that will make the cheating party liable to facing penalty from a judiciary system. There is inherent trust in the judiciary system.
- **Reputation System**: A reputation system lists the Seller, and any complaints against it. A potential buyer can look up the Seller's reputation, and make the payment first. Further malicious behavior from the Seller is disincentivized through the risk of damage to reputation. There is inherent trust in the reputation system.

By Buyer and Seller's Dilemma, we are referring to solving the problem of trust between the Seller and Buyer of a digital good without involving a third party. The name is coined by analogy with the well-known Prisoner's Dilemma [12], as the participants in this game must also think about whether to cooperate (i.e., act honestly) or defect (i.e., cheat on each other). However, in this case the game that emerges is a sequential or extensive-form game because the Seller and Buyer don't move simultaneously.

B. Trust, Enforceability, and Incentive

In order to remove trusted third parties, there must be constraints that are enforced on the behavior of two participating actors. Blockchain systems provide a programmatic way to process and regulate transactions that participants propose in an autonomous manner, referred to as smart contracts [13]. It is important to note that in the context of implementing a protocol using a blockchain-based smart contract, only those interactions which happen through the smart contract can be verified, policed or constrained. Interactions that occur outside the smart contract — off-chain interactions between sellers and buyers — cannot be constrained.

Therefore the only way to influence the off-chain behavior of the participants is by providing on-chain incentives to the actors to conform to good behavior. The crux of our protocol, which we refer to as the dual-deposit escrow trade protocol, lies in the design of such an incentive scheme. We further conduct game theoretic analysis of this incentive scheme, and prove that it is in the best interest of both parties to display honest, mutually helpful behavior.

II. Solution to the Buyer and Seller's Dilemma

A. Assumptions

- We assume the Product being traded as any digital or physical asset that can be secured against unauthorized use through a digital key. We assume that the Product is accessible only using the digital key d (or is the digital data d itself). We can now use d and Product interchangeably.
- We also assume that the Buyer knows the hashed value $h(d)$ for the Product d, which he/she could use to

[1] A fascinating historical example of dual-deposit escrow, though, is mentioned in Julius Caesar's autobiographical *Gallic Wars* (Book 2) which describes mutual hostage exchange between certain tribes as a form of diplomacy: " all the Belgae... were entering into a confederacy against the Roman people, and giving hostages to one another."

verify that a received product is correct. This is a major assumption, and the fundamental problem of verifying that a given hash actually corresponds to the data of interest to the Buyer should be investigated further. For example, a Merkle root corresponding to the data can be used, and the Buyer can challenge the Seller to prove inclusion of some randomly chosen segment of the data.

- The Buyer knows the address of the escrow contract (which is deployed by the Seller as a first step in the protocol), which contains the advertised selling price set by the Seller.
- The Seller and Buyer both have an asymmetric key pair, with their public keys known to each other. We denote the public/private keypair of the Seller by s_{pub}/s_{pri} and the Buyer by b_{pub}/b_{pri}.
- When the Seller and Buyer interact with the smart contract, the smart contract becomes aware of the addresses and associated public keys of both parties.
- We assume that transaction fees associated with deploying the smart contract and sending transactions to it are negligible compared to the price of the product. In all steps, the party which initializes the step by sending a transaction to the smart contract will pay the gas cost. In some cases where a party is paying gas to successfully challenge malice of the other party, the challenger receives reimbursement for the incurred gas cost (in the last step of the protocol).

B. The Dual-Deposit Escrow Trade Protocol

1) **Seller Deployment:** For each sale, the Seller will publish a new smart contract that includes

   P_d : Price of the Product
   $h(d)$: Hash of the Product
   ID : Contract Nonce

   The Seller must also make a Seller Deposit $\mathcal{E}_S$ to the Smart Contract, which is later refunded.
   ID, the contract nonce, is a unique, one-time number (say, the hash of the contract) generated by the Seller to prevent the Buyer from conducting replay attacks in later stages of the protocol.
2) **Buyer Initialization:** The Buyer then initiates the Smart Contract. The Buyer must pay the price P_d for the product and also make a Buyer Deposit $\mathcal{E}_B$, that is later refunded.
3) **Delivery:** The Seller sends an encrypted version of d, namely, $enc_{b_{pub}}(enc_{s_{pri}}(d, ID))$, to the interested Buyer, possibly on an off-chain channel.
4) **Accept/Reject Delivery:** The Buyer decrypts the data d, then hashes it to check if it matches the previously known $h(d)$. The Buyer then provides a response to the Smart Contract; in this response, it either:
   - Accepts delivery of the Product.
   - Rejects delivery, claims that the Seller has cheated, and tries to prove it by sending $enc_{s_{pri}}(d, ID)$ to the Smart Contract.
5) **Reconciliation:** This step is undertaken by the Smart Contract after hearing from the Buyer in the previous step.
   - In case of Acceptance: Both the Seller Deposit and the Buyer Deposit are refunded to corresponding parties. The Seller also receives the price P_d for the product.
   - In case of Complaint: The Smart Contract will decrypt $enc_{s_{pri}}(d, ID)$ using the Seller's public key s_{pub}. If the Buyer submits a garbage string, then the Smart Contract will slash both deposits, along with the payment. Then it first compares ID to ensure that the ciphertext was corresponding to this transaction. It then hashes d and find $h(d)$, which is then compared with the one that the Seller uploaded while generating the smart contract.
     – If a mismatch is found, then the Seller has cheated, and loses its deposit $\mathcal{E}_S$: it is used to pay for the gas consumed in this reconciliation step, and the rest is slashed (burned). The Buyer gets back its deposit as well as the payment for the product.
     – If the hashes match, then the Buyer made a frivolous complaint, and loses its deposit $\mathcal{E}_B$: it is used to pay for the gas consumed in this reconciliation step, the Product payment P_d is sent to the Seller, and the rest is slashed (burned).

C. Game Theoretic Analysis

The dynamics of this interaction between the Seller and Buyer can be modeled as an extensive form game, with the Seller playing the first move (Step 3 in the protocol description), and the Buyer playing the second move (Step 4 in the protocol description). We analyze this extensive form game to find its Sub-game Perfect Nash Equilibrium (SPNE), the strategy profile for both players that ensures no one has an incentive to deviate in any sub-game of the original game [14].

We use the following labels in the game tree:

N, N' : Non-fraudulent (honest) behavior by the Seller and Buyer, respectively.
F, F' : Falsified data submission by the Seller and Buyer, respectively. For the Seller this would correspond to sending the wrong data, but signed by it's key, in Step 3. For the Buyer it would correspond to trying to dispute the transaction with a replay attack in Step 4.
G, G' : Garbage data submission by the Seller and Buyer, respectively. This corresponds to the Seller sending a string that cannot be decrypted with the corresponding public key in Step 3, or the Buyer doing so in Step 4.
S : Frivolous complaint by the Buyer. This corresponds to disputing while providing evidence of honest delivery.
R : No response by the Buyer in Step 4

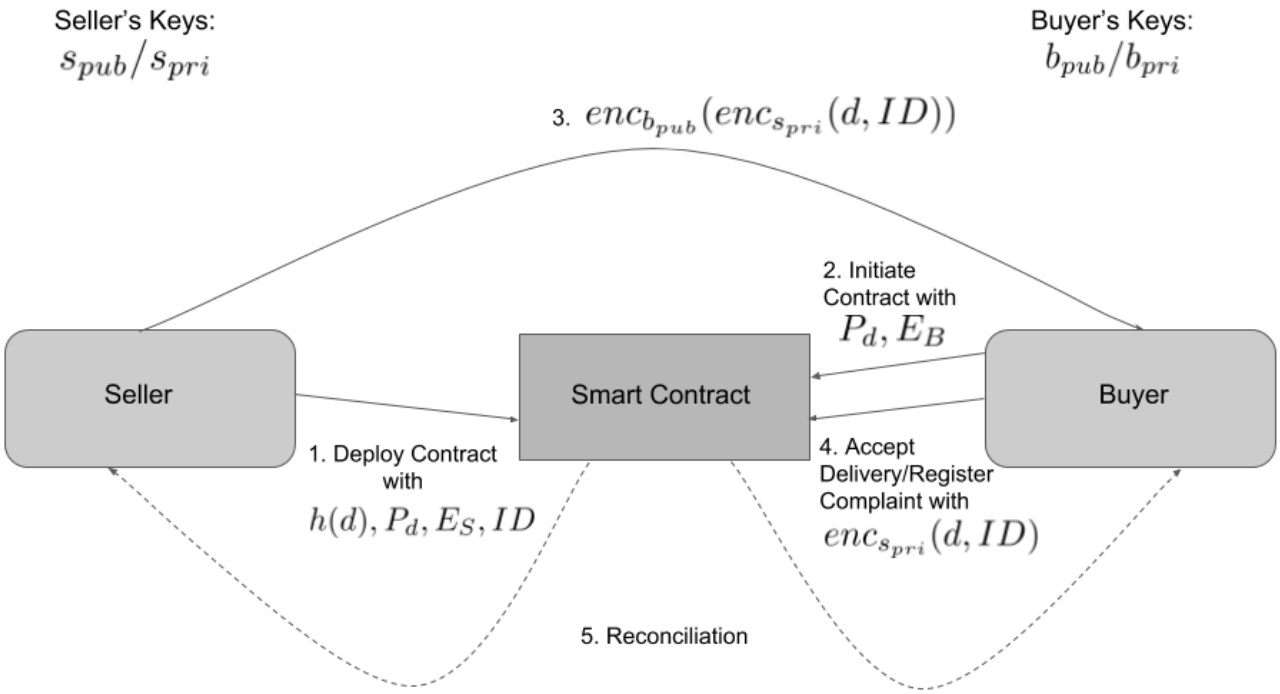

Fig. 1. Proposed System Architecture

Also, we use V_d to denote the perceived value of the Product (from the Buyer's perspective), and assume that $V_d \geq P_d$. We make this distinction to reason about the Buyer's payoff in cases where it receives the Product successfully (and therefore receiving goods of perceived value V_d), but ends up paying an amount different than P_d due to incurred penalties.

We first present the analysis of the payoffs for different interactions between the Buyer and the Seller:

- If the Buyer falsifies its response (i.e., plays F') in Step 4, then regardless of the seller's actions, the Smart Contract will send the payment P_d to the seller and slash the Buyer's deposit E_B. In the case that the Seller was honest (played N), then the Buyer receives the Product, and its payoff increases by V_d.
- Similarly, if the Buyer submits a garbage string (plays G'), then the Smart Contract will slash both parties' deposit, E_B and E_S, as well as the buyer's payment of P_d. In the case that the Seller was honest (played N), then the Buyer receives the Product, and its payoff increases by V_d.
- In the case that the Buyer does not submit any acceptance or complaint in Step 4 (plays R), then the deposits of both parties' and the Buyer's payment of P_d remain permanently locked. These can be treated as a loss in the payoffs. In the case that the Seller was honest (played N), then the Buyer receives the Product, and its payoff increases by V_d.
- If the Buyer is being honest (plays N'), then:
  - If the Seller sent falsified data (plays F), then the Smart Contract can identify cheating on the Seller's part. The Seller's deposit is slashed and the Buyer is refunded it's deposit and payment.
  - If the Seller sent garbage to the Buyer (plays G), then the Buyer sends a garbage string to the Smart Contract. This results in the slashing of deposits and the payment.
  - If the Seller sent the actual data (plays N), then the Buyer accepts the delivery. This results in the payment to the Seller, and refund of respective deposits.

The resulting extensive form game with these payoffs is shown in Fig. 2. Each leaf node of the figure mentions a pair (x, y), where x is the payoff of the Seller, and y of the Buyer, if the strategy profile from the root to that particular leaf is played. It can be determined from backward induction analysis on this tree that there is only one SPNE of $(P_d, V_d - P_d)$ that is achieved by the strategy profile (N, N'), when both parties are non-fraudulent (i.e. honest), so long as $\mathcal{E}_B, \mathcal{E}_S > 0$

D. Safety and Liveness

Safety: The presence of a unique SPNE with positive payoffs for both players guarantees the safety of the protocol. Further, by making the parameters $\mathcal{E}_S$ and $\mathcal{E}_B$ arbitrarily large, we can strengthen the disincentive for malicious behavior to the required standard.

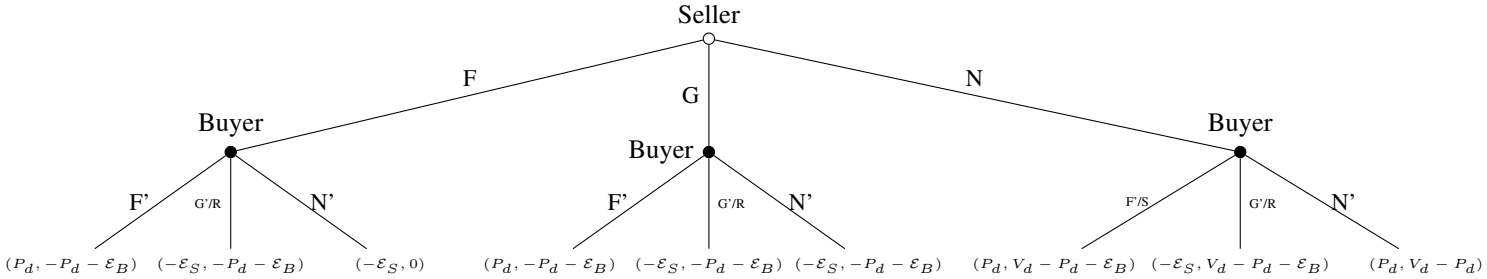

Fig. 2. Game Tree for the Dual-Deposit Escrow Trading Protocol

Liveness: The players are incentivized to move forward in the trade because of the opportunity cost of the large deposit amounts locked in the smart contract. Also, we can guarantee liveness in between Step 1 and Step 2 by making a provision for the Seller to cancel the trade and refund its deposit if the buyer has not moved.

III. Discussion and Future Work

We presented a dual-deposit escrow trade protocol for cheat-proof transactions of payment and delivery between the two participants in the trade of a verifiable digital good. We base our cheat-proof guarantees on a game theoretic analysis of the interactions between the Seller, Buyer and Smart Contract. The safety and liveness properties of the protocol can be improved by increasing the deposit amounts.

While our analysis is suitable for rational but selfish participants, we should note that it does not explicitly cover the case of irrationally malicious parties that are willing to take a negative payoff in order to inflict harm on the other party (although increased deposit amounts may provide some mitigation against such behavior at the cost of raising the barrier to transaction).

We are currently in the process of implementing the proposed protocol as a smart contract on Ethereum. One challenge in such an implementation is the need for decryption in case the buyer disputes delivery; this is not a function that is expected to be used given the analysis, however it is needed for correctness of the protocol; while in theory any Turing-complete smart contract language would allow the implementation of decryption, in practice the use of an oracle might be needed for cost reasons.

A central assumption made in this work is that the digital good being exchanged is verifiable, in particular that the buyer (or the smart contract in case of a dispute, when presented with the relevant evidence) has the ability to verify that the correct good is received — for ease of exposition we assume this is accomplished by a hash of the digital good being known to the buyer and smart contract in advance of the transaction. The problem would be become considerably harder (if not impossible) to achieve without a trusted third party if the buyer cannot independently verify the delivery. Further, we assume that the price of the good being exchanged is already known to both parties. Problems with asymmetric information about prices fall under the broader class of "lemon market" problems [15]. The implications of such lemon market problems in crypto-economic environments is worthy of further study.

The mechanism presented here to pay for the delivery of digital goods could also in principle be used to pay for physical goods provided by the seller that are kept locked in a box that is secured by a digital key [16]. This key could be the digital good in our description with everything else remaining the same. Once the buyer gets this key he/she can open the box to retrieve the purchased physical good (assuming that the hash of the key that is assumed to be known to both parties was generated after ensuring that the physical good is inside the locked box).

We had earlier mentioned that solutions for atomic swap are not applicable to general payment for digital goods, however, the reverse may not be true. We are currently exploring how our approach may be potentially used for the atomic swap problem.

References

[1] P. Timmers, *Electronic commerce*. John Wiley & Sons, Inc., 1999.

[2] S. Goldfeder, J. Bonneau, R. Gennaro, and A. Narayanan, "Escrow protocols for cryptocurrencies: How to buy physical goods using bitcoin," in *Financial Cryptography*, 2017.

[3] L. M. Ponte and T. D. Cavenagh, *Cyberjustice: Online dispute resolution (ODR) for e-commerce*. Pearson/Prentice Hall, 2005.

[4] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," in *The Economics of the Internet and E-commerce*. Emerald Group Publishing Limited, 2002, pp. 127–157.

[5] C. Herley and D. Florêncio, "Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy," in *Economics of information security and privacy*. Springer, 2010, pp. 33–53.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] J. R. Wilson, "Lease security deposits," *Colum. L. Rev.*, vol. 34, p. 426, 1934.

[8] Bit-Bay Market double Deposit Escrow. [Online]. Available: https://bitbay.market/double-deposit-escrow

[9] D. Zimbeck. (2014) Two party double deposit trustless escrow in cryptographic networks and bitcoin. [Online]. Available: http://www.bithalo.org/whitepaper_twosided.pdf

[10] G. Bigi, A. Bracciali, G. Meacci, and E. Tuosto, "Validation of decentralised smart contracts through game theory and formal methods," in *Programming Languages with Applications to Biology and Security*. Springer, 2015, pp. 142–161.

[11] TierNolan. (2013) Atomic cross-chain trading. [Online]. Available: https://en.bitcoin.it/wiki/Atomic_cross-chain_trading

[12] A. Rapoport, A. M. Chammah, and C. J. Orwant, *Prisoner's dilemma: A study in conflict and cooperation*. University of Michigan press, 1965, vol. 165.

[13] V. Buterin *et al.* (2014) A next-generation smart contract and decentralized application platform. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper
[14] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, vol. 393, no. 12, p. 80, 1991.
[15] G. A. Akerlof, "The market for lemons: Quality uncertainty and the market mechanism," in *Uncertainty in Economics*. Elsevier, 1978, pp. 235–251.
[16] T. Buettgenbach and F. Sheehy, "Methods and systems for the physical delivery of goods ordered through an electronic network," Mar. 14 2002, uS Patent App. 09/836,455.