PAR: Payment for Anonymous Routing

Elli Androulaki[1], Mariana Raykova[1], Shreyas Srivatsan[1],
Angelos Stavrou[2], and Steven M. Bellovin[1]

[1] Columbia University {elli,mariana,ss3249,smb}@cs.columbia.edu
[2] George Mason University astavrou@gmu.edu

Abstract. Despite the growth of the Internet and the increasing concern for privacy of online communications, current deployments of anonymization networks depend on a very small set of nodes that volunteer their bandwidth. We believe that the main reason is not disbelief in their ability to protect anonymity, but rather the practical limitations in bandwidth and latency that stem from limited participation. This limited participation, in turn, is due to a lack of incentives to participate. We propose providing economic incentives, which historically have worked very well.

In this paper, we demonstrate a payment scheme that can be used to compensate nodes which provide anonymity in Tor, an existing onion routing, anonymizing network. We show that current anonymous payment schemes are not suitable and introduce a hybrid payment system based on a combination of the Peppercoin Micropayment system and a new type of "one use" electronic cash. Our system claims to maintain users' anonymity, although payment techniques mentioned previously – when adopted individually – provably fail.

1 Introduction

Anonymous networking has been known since 1981 [1]. A more practical scheme, Onion Routing, was first described in 1995 [2]. Currently there is little practical use of network anonymity systems. Some of the problem is undoubtedly sociological: most people do not feel the need to protect their privacy that way; this is one reason that companies such as Zero Knowledge Systems [3, 4] and Digicash [5] failed. Another problem, though, is that strong anonymity against traffic analysis requires cooperation by and implicit trust in many different parties. Any single entity, no matter how trustworthy it appears, can be subverted, whether by technical means, corrupt personnel, or so-called "subpoena attacks". All known solutions require, and in fact enforce, routing through multiple parties. This, though, introduces another problem: economic incentives. In a single-provider anonymity scheme, that problem is conceptually simple: the party desiring privacy pays a privacy provider. This payment

can be protected by digital cash [6]. Unfortunately, in a multi-provider Mixnet or onion routing network, the problem is more complex, since each party must be paid. By examining existing digital cash schemes, we show that they do not provide the necessary cost or privacy properties required to maintain anonymity. For example, in Chaum's original e-cash scheme [6] a double-spender's identity is exposed. This is perfectly acceptable – double-spending is a form of cheating that should be punished – but in the context of an onion routing network, detecting double spending gives an adversary clues to path setup.

To address these problems, we propose a novel hybrid payment scheme by combining features from Micali's micropayment system [7] and a light-weight, blind signature-based e-cash scheme. Our goal is to create incentives for the network participants to act in a cooperative manner based on their personal interests. We show that any solution must be sound in several dimensions. First, it must protect privacy. This is not trivial; witness the many (partial) attacks on various anonymous networking protocols [8, 9]. That said, we do not claim to have fixed those problems. Rather, our aim is avoid introducing any new vulnerabilities that stem from the payments scheme.

Second, we want a system that is in principle deployable. That is, though we assume such things as anonymous payment systems, we do not assume, for example, incorruptible banks. More importantly, we want a system that is compatible with known economic behavior. Therefore, while our system assumes that people are willing to pay for privacy, we want a system where customer payment – the profits of forwarding nodes – are related to privacy desired and effort expended. In essence, there must be a profit motive and the opportunity for market forces to work. To deter exploitation of the payment scheme, we provide mechanisms to detect cheaters: those parties who accept payment but do not provide services.

Third, we do not attempt to achieve absolute financial security. Instead, we are willing to accept small amounts of cheating, by senders or forwarders, as long as the amount is bounded and limited (possibly with some trade-off) by the party who is exposed to loss. Finally, we want a system that is acceptably efficient in practice and does not impose unreasonable resource consumption. To that end, we evaluate the operations of a prototype PAR – which stands for Payment for Anonymous Routing. Our initial performance evaluation indicates that PAR is highly configurable and can operate with acceptable communication and CPU overhead. As opposed to previous work on incentivised anonymity, which

used mixnets ([10], [11], [12]), our system guarantees usable efficiency, accountability and maintains anonymity against traffic analysis attacks.

2 System Considerations

We will examine current anonymizing networks and payment schemes and show why current payment schemes, when applied to onion routing schemes, fail to maintain anonymizing network properties, while our hybrid scheme succeeds. Furthermore, we set up the threat model and we identify the individual components and the properties required by a payment scheme to provide the same protection the network anonymity system was designed for.

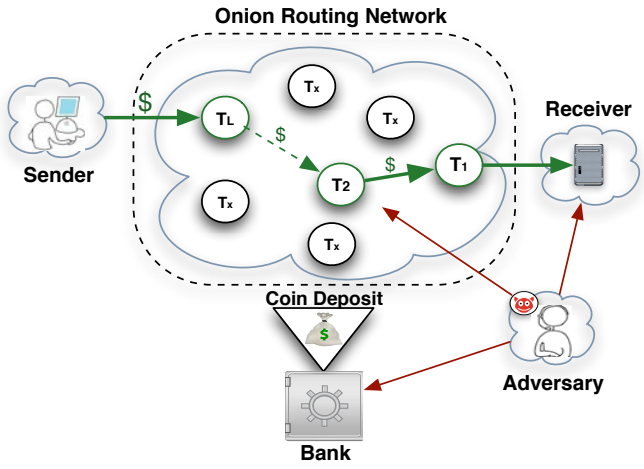

Fig. 1. The PAR architecture combines an onion routing anonymity network (Tor) with a payment scheme. Each node $T_1, T_2, T_3, \cdots, T_L$, where L is the path length, in the path from the sender to the receiver receives payment in coins for its service.

Anonymizing Network. An anonymizing network is a particular type of peer-to-peer network, in which peers communicate anonymously. Anonymizing networks aim to offer sender anonymity even against the recipient as well as sender-receiver unlinkability. Neither the recipient nor any other participant should be able to detect the actual sender with a better probability than selecting the sender at random. As a proof of concept, we use Tor [13], the second generation onion routing anonymity network, a well-known and deployed network anonymity system.

Adversarial Model. The participating entities of our system are the Tor relays, the outside users, and a clearance entity, *i.e.* a Bank, where monetary units are deposited/withdrawn. We inherit Tor's local adversary model where users can only observe the traffic going through them and a limited amount of the rest of the network traffic. In addition, we assume that malicious users can manipulate any packet going through them and use this information to compromise anonymity. The Bank, on the other hand is assumed "honest but curious". Therefore, although trusted to be honest in all of its functional operations – cash withdraw and deposit – the Bank can collaborate with any number of users in order to disclose the initiator of a communication or active communication paths. We do not consider covert channels for anonymous communication with routers without paying as a part of our threat model.

System Requirements. Our primary requirement is that the overall system should maintain the anonymity provided by Tor even when the payment deposit information is exposed to a third party including the Bank. Anonymity, however, should not be achieved at the expense of efficiency. Moreover, the payment scheme should meet the requirement necessary for any payment system such as accountability, correctness, and robustness.

Payment Analysis. For our analysis, we classify current payment schemes in two categories: *Identity-bound payments* and *Anonymous payments*. In Figure 1, the sender provides payment for all nodes T_1, T_2, T_3, $\cdots$, T_L[3] that forward the sender's traffic to the receiver. We will show that both of the current payment schemes, when applied to a Tor network, render the anonymity system vulnerable to attacks that compromise the anonymity of the senders.

Identity-bound Payment Schemes. Identity-bound payments constitute signed endorsements from the payer to the payee. Accountability and robustness are the two main features of this class. The micropayment scheme [7] is an example of an Identity-bound payment. It was designed to be efficient for small, online transactions. When used to pay Tor nodes, identity-bound payments provide immediate accountability because invalid payments from any entity can be easily accounted for. However, when applied in the context of the Tor network, this property has adverse implications: upon clearance, the Bank obtains global knowledge about all transactions in the anonymity network. If the sender uses his own coins to pay the nodes in the path, his identity is exposed to

[3] In Tor, intermediate communication path nodes are chosen randomly by the communication initiator.

them. Therefore, any node in the path to the receiver can identify him with the help of the Bank. To make things worse, the last node in the path – who may suspect that he is the last node if the receiver is outside Tor – can link the sender to the receiver. A potential way to work around this problem is to distribute payments only to immediate neighbors. With this payment strategy, the sender pays T_L with L coins, T_L pays T_{L-1} with $L-1$ coins etc. This approach makes path tracing much harder and leaks less information but it is far from secure: deposits made by the sender to the first Tor node are still available to the Bank. Counting the coins bound to the sender's identity, the Bank can infer with high confidence the number of packets communicated to the sender and link the sender to the receiver. This analysis indicates that having identity-bound coins reveals too much information, enabling an adversary with access to payment information to break the system's anonymity using simple inference techniques.

Anonymous Payment Schemes. In this scheme, the payment does not carry any identification information of its initial owner. Chaum's Digital cash [6] and the later versions [14–16] of Tunstall et al. and Camenisch et al. are perfect examples of such anonymous payment schemes. In the general case of digital cash systems, a user withdraws money from a Bank, which he can only spend himself and which when legally spent can never be linked to his identity. Merchants deposit the coins they have received to check whether any of them has been spent more times than its nominal value (double-spending). If the later occurs, the identity of the double-spender is revealed. However, all the anonymous payment schemes demand excessive communication overhead for each transaction because there are a lot of messages that need to be exchanged between the sender and the path nodes.[4] This requirement makes e-cash schemes impractical for our system.

An alternative solution would be for all users to withdraw a special kind of anonymous coin from the Bank, which can simply be Bank blind endorsements [17], and use these coins to pay the intermediate Tor nodes. Ideal as it might initially seem, using a completely anonymous payment scheme with Tor has its drawbacks. First of all, there is no immediate accountability, since double-spending in this case will not reveal the double-spender. Thus, to prevent double-spending, any payments received should be immediately checked and deposited in the Bank. Unfortunately, im-

[4] In the compact e-cash payment scheme [16], which is considered efficient a single "spend" procedure in e-cash systems would requires at least two rounds of message exchange between the sender and every node in the path.

mediate coin deposits could lead to deposit timing attacks exposing Tor's anonymity. More specifically, the timing of deposits by the nodes along a Tor path discloses to the Bank the path as well as an estimated of the number of packets transferred. Accumulating deposits for appropriately long time intervals – sufficiently long that many connections are established, to mitigate timing attacks – would increase the amount of unchecked coins and thus of double-spending. Indeed, since anonymous coins are not traceable beyond the first Tor node, sending valid coins only to the first node is enough to prevent it from been traced. For the rest of the nodes, the cheater uses double-spent coins, exploiting this deposit strategy by transmitting many packets in a short period of time.

Our Contribution: Hybrid Approach. Both of the two aforementioned classes of payment schemes have advantages and disadvantages. Our approach creates a hybrid payment scheme by combining the two payments methods into a single one. In particular, nodes outside the anonymizing network withdraw an initial number of anonymous coins (A-mcoins) from the Bank and use them to pay the first node in the Tor-path (T_L) they have chosen. T_L then uses micropayments[5] to pay T_{L-1}, who also uses micropayments to pay its neighbor. Each time, the amount of money paid decreases according to each node's price. Nodes participating in the Tor network follow the same protocol with the option to use either anonymous or micropayments for the first node in their forwarding path.

In addition, each of the payment coins in the scheme has a corresponding receipt and becomes valid only when it is submitted for deposit together with the receipt. As we will show in the following sections, our payment scheme combines all the desirable properties of the existing payment schemes, but without maintaining any of the problem each one of them causes when used individually and in this way it provides sender-receiver unlinkability along with accountability and efficiency.

3 High-Level Description of PAR Protocol

Here we provide a high-level description of our payment scheme. To help the reader, we start with a brief description of the Tor circuit setup; we then present our payment scheme.

3.1 Tor

Tor is formed by a set of relay nodes (onion routers) that act as traffic indirection points. The region in the dotted lines in Figure 1 de-

[5] Identity-bound payment

picts a typical communication in Tor. Each onion router maintains a TLS [18] connection to every other onion router. To establish communication, the sender selects a random sequence of Tor relays to form a path to the receiver or what is called a circuit. In Figure 1, the sender selected nodes $T_1, T_2, T_3, \cdots T_L$, where L is the path length. The sender constructs circuits incrementally, by negotiating a symmetric key with each onion router on the path, one hop at a time. Initially, the sender contacts the first path node, T_L, and they both commit in a Diffie Hellman (DH) key agreement procedure. Once this initial circuit has been created, sender uses T_L to extend the circuit to T_{L-1}. In particular, T_L and T_{L-1} establish a circuit – through the TLS channel they share – which T_L relates to the one with the sender. Sender commits anonymously (using T_L as mediator) in a Diffie Hellman (DH) key exchange procedure with T_{L-1}. Repeating this process through the extended tunnel, the sender may add more Tor nodes to the circuit. At the final stage, the last node in the path, T_1, opens a data stream with the receiver and a regular TCP connection is established between the sender and the remote site's IP address. At the end of the circuit setup procedure, every relay in the path shares a secret key with the anonymous path initiator, as well as with each of his path neighbors. The key a path node shares with each of his neighbors is only used for securing their part in the communication path. Each transmitted Tor message along a path, contains an unencrypted header with a circuit ID and a multiply-encrypted payload. At each hop, the corresponding path node decrypts the payload – using the key that node and the sender share – and replaces the circuit ID with the one that corresponds to his circuit with next node in the path.

3.2 PAR

We introduce the hybrid payment scheme from the previous section to the Tor network; again, see Figure 1. In our scheme, payments are conducted between consecutive nodes on the forwarding paths and added inside the transmitted messages using an additional encryption layer. Each forwarding node T_i creates payment coins for its path successor T_{i-1} using sender S's directions and adds these payment coins to the onion message to be forwarded to T_{i-1}. Payment information is provided to each T_i through the secret channel it and the sender share. To avoid exposure as in Tor, T_i further encrypts the resulting message with the key it shares with its successor. To complete the payment transaction and for the coins to become valid, every relay node has to receive the receipts for its payment by

its successor. Therefore, each node, other than the last one, upon validating the received message, sends to its predecessor the payment receipt. S controls the payments made along the forwarding path by supplying the receipts for all the coins used.

To avoid cheating, S provides each path node T_i with additional information for it to verify that the payment received from T_{i+1} is indeed valid. Receipts are forwarded to T_{i+1} if and only if the the payments are valid. Since the circuit is used in both directions (*i.e.* to both receive and transmit messages, the last node can either be pre-paid or paid after the delivery of the message by the sender depending on the acceptable bounded risk. In either approach misbehaving nodes will be detected within the first round of sent messages and will be excluded from the forwarding path, which will cause them more loss than the expected gain from fraudulent behavior and they will have no incentive for cheating.

The initial setup stage for Tor circuits will be extended with nodes sharing some hash function that will be used prevent third party manipulations in the payment protocol.

4 A Hybrid Payment Scheme

In this section, we present a detailed description of our payment protocol. However, before proceeding, we first define three properties required to preserve anonymity in an onion routing network:

Sender-Receiver Unlinkability. Let S be a user, who may or may not be a member of the anonymizing network, who sends a message M anonymously[6] to a user R. Then nobody except a global adversary, even with the collaboration of a third party and R, should be able to link sender and receiver or reveal the path between them.

Usable Efficiency. This refers to the fact that the overhead in the packet exchange for the payment scheme and the CPU overload with additional cryptographic operations will be reasonable and will not impede the normal functioning of the system.

Accountability. This property ensures that any cheating node trying to forge messages or double-spend coins is caught and expelled from the network.

4.1 Payment Coins

We use two types of payments that consist of two parts: a payment part, which we will call a coin, and a receipt part. A coin becomes valid only

[6] Here, "anonymously" means "using the anonymizing network".

when it is accompanied by the corresponding receipt. The receipt is a random number that is bound to the coin by incorporating its hash value in the coin. Thus a random number r serves as a receipt for the coin that contains the hash $H(r)$. Although similar in structure, the two types of payments have different properties and that is why they are named differently: micro-coins (*S-coins*) and anonymous coins (*A-coins*).

S-coins(Signed microcoins). S-coins are generated and used for payments between Tor participants. They are based on the micropayments introduced in [7] but with the addition of receipts. An S-coin is an extension of a microcoin MC :

$$SC_{T_i \to T_j} = sig_{T_i}\{MC, H(r), T_j\}.$$

As in the microcoin case, an S-coin is strongly bound to both the identity of the node T_i, who generates it by signing its content, and the identity of the payee T_j. Finally, it contains the hash of the receipt $H(r)$ that makes the coin valid. The microcoin part of the S-coin MC contains the transaction details τ as well as a sequence number – according to micropayment scheme [7] – without containing any timing information.

S-coins inherit the properties of microcoins. Only a predetermined fraction of them are payable, while no participants in the payment scheme can find out in advance which coins will become payable.

A-coins (Anonymous coins). A-coins use the idea of e-cash ([6]). They are generated by the Bank upon users' requests. Users outside Tor buy a predetermined number of A-coins from the Bank and pay with them for using the anonymizing network. Members of Tor also acquire a number of A-coins and may also use them. All A-coins are of the form

$$AC(r) = sig_B\{r\},$$

where r is a random number generated by the User, and $sig_B\{r\}$ is the blind signature of the Bank of r. A-coins are all payable and subjected to double-spending checks.

4.2 Payment Protocol

Figure 2 presents in detail the messages exchanged in the payment protocol. We further analyze the individual protocol stages.

Initial Set-up All nodes participating in Tor acquire a public-private signature key pair (sk_U^s, pk_U^s) and a public-private encryption key pair (sk_U^e, pk_U^e), used to interact with the other members in the network.

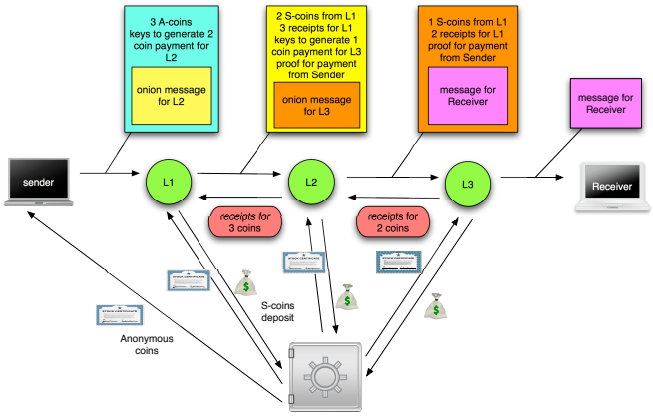

Fig. 2. The intuition behind our payment protocol is that Tor participants use S-coins to avoid exposing the forwarding path; outside senders, by contrast, use A-coins to maintain their anonymity.

Bank generates a blind signature key pair (sk_B^b, pk_B^b) for signing A-coins. In addition to the hash H already used in Tor for integrity purposes, we establish another collision resistant hash function H_r for the coins' receipts. At the end of the circuit setup procedure in Tor, the sender shares with each node T_i in the path a secret key K_{ST_i} while any two consecutive nodes in a path share a secret key $K_{T_iT_{i+1}}$. In our system, the sender agrees with each path node on a hash function H_{ST_i}. The shared keys are used for communication encryption whereas the hash functions for integrity checks. We use M_k to denote message M encrypted under key K; $sig_U M$ is the signature of user U on M.

Payment Generation A-coins are generated in cooperation with the Bank. When user U wants to obtain A-coins for payment, he generates a fixed set of random numbers $r_1, r_2, \ldots, r_n$, which serve as the receipts for the coins. Then, the user submits to the Bank the hashes $H_r(r_1), H_r(r_2), \ldots, H_r(r_n)$ which in turn signs them and generates coins of the form:

$$AC_i = sig_B(H_r(r_i)).$$

The resulting A-coins can be used for payment to any node in the network.

In the case of S-coins, users can generate them but they have to specify the payee. When user U wants to pay a node T_i with an S-coin, he

generates the random number receipt r and its microcoin-like part MC which consists of a number that increases by one per S-coin payed by U to T_i and no timing information at all. The final form of the S-coin is:

$$SC_{U \to T_i}(r) = sig_U(MC, H_r(r), T_i).$$

Communication Protocol Description Let S send to R a message M through the path $T_\ell, \ldots, T_1$. The following sequence of payments occurs for the transfer of the message:

- S pays T_ℓ ℓ coins, which may be A-coins or S-coins. Nodes outside Tor can only pay by A-coins while Tor nodes can use either type of coin.
- each node T_{i+1} on the forwarding path pays its successor T_i i S-coins.

The sender S chooses the receipts that will be used by the nodes on the path to generate payments for their successors. It also sends proofs to each of the nodes T_i in the form $H_r(r_1), \ldots, H_r(r_i)$ where $r_1, \ldots, r_i$ will be the receipts for the coins the node will get from its predecessor.

A node T_{i+1} gets the receipt for its payment coins from its successor T_i on the path.

Exchanged Messages The general form of the message that a node T_{i+1} sends to a node T_i on the forwarding path between sender and receiver is the following:

$$(\ \{T_i,\ \text{coins for } T_i,\ sig_{T_{i+1}}\{H(\text{coins for } T_i)\},\ \{M_{S \to T_i}\}_{K_{ST_i}}\}_{K_{T_{i+1}T_i}}\ )$$

- T_i specifies the receiver of the message
- "coins for T_i" is the payment the node gets for forwarding the packet. The coins here are either A-coins if the sender was an outside node and T_i is the first node in the path, or S-coins of the form $SC_{T_{i+1} \to T_i}$
- $sig_{T_{i+1}}\{H(\text{coins for } T_i)\}$ is mainly needed in the case of A-coins[7] and serves accountability purposes when double-spending has been detected and
- $\{M_{S \to T_i}\}_{K_{ST_i}}$ is the part of the onion message from the sender that has to be read by T_i.

Now consider the last part of the message $M_{S \to T_i}$, which has the following form:

$$(\ T_{i-1},\ T_{i+1}\ \text{receipt},\ \text{payment guarantee for } T_i,$$

[7] it can be eliminated in the case of S-coins

values for generation of coins for T_{i-1}, $\{M_{S\to T_{i-1}}\}_{K_{ST_{i-1}}}$)

- T_{i-1} is the successor of T_i on the path
- the receipts for T_{i+1} are the random numbers that the sender generated encrypted with the key $K_{ST_{i+1}}$; T_i sends them back to its predecessor on the path
- the guarantees that T_i receives for its payment are of the form: $H_{ST_i}(r_1), \ldots, H_{ST_i}(r_j)$, where $r_1, \ldots, r_i$ will be the receipts for the coins he was paid with
- $\{M_{S\to T_{i-1}}\}_{K_{ST_{i-1}}}$ is the part of the onion message from the sender that has to be forwarded to T_{i-1}. In the case when T_i is the last node on the forwarding path, $M_{S\to T_{i-1}}$ is the message to the receiver.

After receiving its message from its predecessor, the node T_i acquires its payment, which is verified using the guarantees received from the sender. Then, it sends the receipts for T_{i+1} to its predecessor. Next, the node uses the values from the sender to generate payment coins for its successor T_{i-1}. It adds the coins to $\{M_{S\to T_{i-1}}\}_{K_{ST_{i-1}}}$, signs the whole resulting message and forwards it to its successor.

Deposit The deposit of all coins is handled by the Bank, which checks their validity and depositability. The validity of S-coins can be checked immediately by each node which is paid with them while the validity of A-coins is established at the Bank that checks for double-spending. At each deposit time the nodes deposit all coins that they have received during the period. Detailed analysis of the deposit period is provided in a later section. Here, we define the procedure for deposit. Coins are considered for deposit if and only if they are accompanied by the corresponding receipt. The valid coins will be handled in two different ways: The deposit of S-coins is, in essence, a deposit of the underlying microcoins. This means that only a fraction of them will become depositable [7]. All A-coins are depositable at their nominal value.

4.3 Discussion

We preserve Tor's anonymity by allowing each node on the path to know only its predecessor and its successor. To this end, we harness the layered structure of the message passed by the sender to the forwarding path and the fact that payments are made between consecutive nodes. However, the sender still has control of the payments made along the path by sending the receipts used for their generation. A node that attempts to cheat can be easily identified by its successor. Since the successor holds the receipts

for the cheater's payment there is no incentive for the cheater to either mangle or drop the message. Finally, Tor encryption guarantees both the confidentiality and integrity of all transmitted messages.

5 Security Analysis

There has been a wealth of research related to attacks against onion routing systems including Tor. Our goal is to ensure that PAR does not introduce new types of attacks, especially ones that can target either the anonymity or the robustness of an onion routing system. In addition, we prove the security properties of PAR using the augmented Tor threat model introduced earlier.

Sender-Receiver Unlinkability and Deposit Rate

We provide a formal model of information leakage of the payment scheme that can expose anonymity when combined with known attacks against anonymity networks. Although two differentiable types of payments are used in PAR this does not bring any higher risk than currently exists in Tor for the identity of the senders, which can be recognized as such if they use A-coins. The reason for this is that only nodes outside the system are required to pay the first node in their forwarding path with A-coins and currently lists of the relay nodes in Tor are publicly available and therefore outside nodes using the anonymizing system can be also recognized by the first relay that they use.

We will consider attacks that have access to the deposit information in addition to corrupted nodes. In our payment scheme, the Bank can be considered a global adversary since it observes the deposits of coins made at all nodes. That is why in the analysis of possible attacks we will speak in terms of whether the Bank can disclose any of the anonymization that occurs in Tor's forwarding paths, with or without cooperation from malicious nodes.

The most serious type of attack for an anonymization network is one that manages to link senders and receivers communicating over the network. Since the senders using PAR pay with anonymous coins if they are outside nodes, the Bank cannot identify the start of the path that they choose to use. If the sender is a Tor node that forwards other traffic as well, the payments for all of its own and forwarded traffic are indistinguishable; hence the Bank cannot trace the traffic originating at the node just by observing deposits. The receivers are also unidentifiable by the Bank, since there is no monetary transaction between the last node and the receiver.

We have shown that the Bank by itself cannot link sender and receiver. Now we must consider the question whether an adversary observing the deposits can obtain partial information about a forwarding path by discovering three consecutive inside nodes in the path, i.e., being able to guess to where a node forwards packets received from a particular predecessor. Consecutive nodes in a path can be inferred from the signed coins deposits, but the only thing that this means is that there is at least one path that has that pair; nothing more is learned about which connection this path serves.

For the purposes our analysis let $cp^{T,\tilde{T},i}_{<T_\ell,\ldots,T_1>}$ be the packets transferred on a connection path such that $T = T_i$ and $\tilde{T} = T_{i-1}$. We denote the packets on all connection paths that have $\tilde{T}$ as a successor of T by

$$C(T,\tilde{T}) = \{cp^{T,\tilde{T},i}_{<T_{i_l},\ldots,T_{i_1}>} | 1 < i \leq \ell\}.$$

Then the number of coins that a node $\tilde{T}$ will receive from T will be

$$G(T,\tilde{T}) = \sum_{\forall cp^{T,\tilde{T},i}_{<T_\ell,\ldots,T_1>} \in C(T,\tilde{T})} i * c^{T,\tilde{T},i}_{<T_\ell,\ldots,T_1>}.$$

If we denote the number of anonymous coins that a node T deposits with $G_{ac}(T)$, we can calculate the number of packets forwarded by T (assuming that a node is paid with one coin for each packet forwarded):

$$\sum_{T'} G(T',T) + G_{ac}(T) - \sum_{T''} G(T,T'').$$

In order to hide the exact number of packets that it has forwarded, a node can deposit some of its own anonymous coins; thus the above expression will no longer be a correct estimate. Not knowing the rate of packet transfer nor the number of connections in which two nodes are consecutive, an adversary cannot receive enough information just from the deposits of coins to determine three consecutive nodes in a path.

Let us now assume that there is a malicious node that colludes with the Bank in order to reveal more about a path. The malicious node can disclose his predecessor and his successor on a particular connection path, as well as his position in that path. Let $T = T_i$ be such a malicious node in the path $T_\ell, \ldots, T_1$. Now the adversary can find out who are the nodes T_{i+1} and T_{i-1} and the number of packets k that T_i forwarded on that connection. The only thing that it can infer about the identities of T_{i+2} and T_{i-2} is that if

$$(i-1) * k > G(T_{i-1},\tilde{T}) \tag{1}$$

then the node $\tilde{T}$ cannot be a successor of T_{i-1} and similarly if

$$(i+1) * k > G(\tilde{T}, T_{i+1}) \tag{2}$$

$\tilde{T}$ cannot be a predecessor of T_{i+1}. This is true only if we assume that the connections among different nodes have the same forwarding rate. Thus the chance of the adversary finding out anything more about the path than what it would have found out from a malicious node in Tor without any payments is very small.

In the discussion above we have made an implicit assumption that the deposits of coins occur at certain intervals during which enough connections have been established. The statement "enough connections" means that there are no cases where only one node deposits another node's signed coins and it is clearly its successor in any connection. Also, we minimize the probability of Eq. 1 or Eq. 2 being true.

Deposit Rate Now we give an estimate of what we consider "enough" connections and packets transferred during a deposit period. The situation in which an adversary may eliminate a link between two Tor nodes as being part of the path transferring the packets on a particular connection is when the payments made for that link are not enough for the packets that were expected to be sent on the connection. To avoid such situation, we want the expected payments made for packets forwarded along a link between any nodes during a deposit period to exceed the expected payment for the packets forwarded on a single connection.

Let us assume that there are N packets sent across a network consisting of n nodes over C connections during a deposit interval. Let L be the average length of the forwarding path. Then since the probability of a node being in any position on the path is $\frac{1}{n}$, the expected payment that a node will get per packet sent over PAR will be

$$\frac{1}{n}(1 + \ldots + L) = \frac{L * (L+1)}{2n}$$

Now considering that every node will forward on average $\frac{N}{n}$ packets, a node will be paid $\frac{N*L*(L+1)}{2n^2}$, which distributed across the $n-1$ edges going out of it yields $\frac{N*L*(L+1)}{2n^2*(n-1)}$ payment per edge. At the same time the average payment made for the packets on a connection is $\frac{N*L*(L+1)}{2C}$.

We observe that for

$$\frac{N * L * (L+1)}{2n^2 * (n-1)} > \frac{N * L * (L+1)}{2C}$$

to hold, we need $O(n^3)$ connections across the whole network or an average of $O(n^2)$ connections per node. We stress that with so many connections, an adversary would not be able to eliminate even a single possible path route for a given connection. If we now consider the situation when the adversary can narrow the possible successors of a particular node down to some number n_c, there are still n_c^ℓ possible paths for the connection. However in this case we would want

$$\frac{N * L * (L+1)}{2n^2 * n_c} > \frac{N * L * (L+1)}{2C}$$

and we will need a total of $O(n^2)$ connections across the network or $O(n)$ per node.

In previous discussion we mentioned that each node may deposit some of its own anonymous coins to provide more anonymity of the traffic it is forwarding. We now point out that by having each node deposit anonymous coins we will additionally disguise the entry points for outside traffic being forwarded in the network. Since the ratio of anonymous and signed coins in the payment scheme is $\frac{2}{L-1}$, to preserve this ratio across all nodes each node should add its own anonymous coins to maintain the same deposit ratio.

Usable Efficiency The efficiency of our payment scheme is comparable to that of micropayments [7, 19]: the majority of the payment coins in our system are signed coins based on microcoins with the additions of receipts. These are much more efficient than ecash [6], which requires zero knowledge proofs. (Even our anonymous coins are lightweight blind signatures.)

Accountability The accountability property requires that the identity of a node that behaves maliciously – double-spending, forging attempts, message manipulation, etc. – will be revealed along with a proof of his guilt.

No node can tamper with the forwarded onion message since it is protected with layers of encryption that can be opened only in the corresponding order. Thus any attempt for forgery will be exposed by its successor. In addition, no double spending is possible for S-coin payments. Each of the coins is a signature by the spender; furthermore, it specifies the receiver and the payment details.

Double spending for anonymous coins is possible and can only be detected at deposit procedure. However, messages containing A-coins, contain also signed hashes of the coins, which serve as proof of A-coins' origin if a double-spending has occurred. Thus, the nodes paid with the same coin have an proof for the misbehavior.

There is an issue of whether maintaining logs of coin related message exchanges is necessary after coins' deposit for satisfying accountability in our system. Indeed, keeping some A-coin/S-coin related logs is required to detect malicious actions by the spender/payee; In particular Bank is required to keep a log of the serial numbers of the A-coins that have been deposited so far and as well as the biggest serial number of S-coins each pair of peers has exchanged. The A-coins exchanges are required to be maintained for detecting the double-spender but only for the time of one deposit period.

Thus far, we have showed that our payment scheme abides by its design principles. We now prove that it still satisfies properties common for any viable payment scheme.

Correctness When all participants act honestly and follow the protocol, our payment scheme fulfills its goals: all packets are delivered, the nodes on the forwarding path are paid, and the anonymity of the sender and receiver is maintained. If all nodes properly forward the onion message that is initiated by the sender it is guaranteed to reach its receiver because each forwarding node knows where exactly to send it. According to the payment scheme, each node receives exactly one coin more than it has to pay its successor per packet. Thus all nodes are paid equally for their service. We have already shown that payments observed by the Bank are not enough to compromise the anonymity of the identities of sender and receiver.

Robustness Robustness refers to the probability that the path chosen by the sender will be secure in the presence of malicious parties in the network. Let us assume that the fraction of malicious nodes is α. Then the probability that there is no malicious node on a path of length l is $(1-\alpha)^l$. The computed probability, however, is important for the case when we assume that a malicious mode on the path prevents the traffic, i.e. it drops or misdirects it. This also holds in Tor with no payments. Now we restrict our attention to malicious nodes only in the context of the payment system, i.e. nodes that may expose the connections going through them and the corresponding payments for them. Based on our analysis showing that a node acting in this malicious way can disclose its predecessor and successor in the forwarding path, at least half of the nodes on a path will have to be malicious in order to expose the identities of sender and receiver. Thus the probability of preserving the anonymity of sender and receiver over a path of length l is $(1-\alpha)^{l/2}$.

Monetary Unforgeability No coin forgery is possible in the payment scheme since both types of coins are protected with signatures.

Signed coins contain personal signatures of the payer; anonymous coins contain the Bank's signatures.

6 System Performance Evaluation

In this section, we quantify the computational overhead added to Tor by our payment scheme. We execute the `openssl speed` command 1000 times and compute the average estimated running time of blind and digital signatures (RSA), and symmetric key encryption and hashes (SHA1). We will focus on the overhead imposed on the communication initiator S as well as on a random path node T_i.

We define c_h to be the cost of a hash function, c_e the cost of a symmetric encryption procedure, and $c_s(c_{bs})$ and $c_{vs}(c_{bvs})$ the (blind) signature and (blind) signature verification cost. For 1024 byte messages hashed with SHA1, $c_h = 0.0045$ milliseconds. For CBC DES encryption[8] in blocks of 256 bytes and RSA signature and verification in blocks of 1024 bytes the estimated running times are $c_e = 0.020$, $c_s = 3.361$, and $c_{sv} = 0.142$ milliseconds. Assume a path of length L. For each payment round, S has to generate L receipts for the required A-mcoins and have them blindly signed by the Bank, and symmetrically encrypt the A-mcoins' receipts with $K_{ST_{L-1}}$. In addition, S should calculate the content of S-mcoins that each path node T_i will pay its successor T_{i-1}, and encrypt the receipts with $K_{ST_{i-1}}$ key. Thus the overall computational cost for S for each payment round would be:

$$Cost_S = L * (c_{bs} + c_h + c_e) + \frac{L * (L-1)}{2} * (c_h + c_e)$$

For the usual case of $L = 4$, $Cost_S$ averages to 14.24 milliseconds overall, or to 1.4 milliseconds per coin to be paid.

On the other hand, each node T_i in the path, should create $i - 1$ for T_{i-1}'s S-mcoins and verify the validity of S-mcoins it received by T_{i+1} (signature verification and receipt):

$$Cost_{T_i} = i * (c_{vs} + c_h) + (i-1) * c_s$$

In this case T_i will have to spend 0.045 milliseconds for each coin it gets payed and 3.36 milliseconds for each coin it pays.

The performance impact of our scheme is dominated by two factors: the path length and the number of packets per payment. However, the two

[8] We used DES for our tests, precisely because it is slower than AES; we wished to set a lower bound on performance.

have very different properties. The number of packets per payment, N, represents the tradeoff between performance and risk. By setting N high, the total cost of our scheme is minimized, since the expense is amortized over a large number of transmissions. However, N also represents how willing nodes are to transmit packets without assurance of payment. If N is too high, a cheater can send a fair amount of data before being caught. Minimizing that risk requires setting N low, and hence increasing the cost.

7 Related Work

Previous research on applying payments in anonymizing networks was focused on mixnets: Franz, et al [10], Figueiredo et al. [11] and Reiter et al. [12] all use a blind signature type of electronic cash to induce mixes to operate honestly. The approach of Franz et al. divides electronic payment and messages into small chunks and allows mixes and users to do the exchange step-by-step, which made the resulting system extremely inefficient. Furthermore, the receiver is required to participate in the payment procedure, which is undesirable: the receiver may not know or care about Tor. Figueiredo provided a completely anonymous payment system for mixnets, but without any accountability and robustness. Reiter et al. proposed a fair exchange protocol for connection-based and message-based mixnets. However, their protocol assumes that mixes would work properly to receive their payment after they commit to their service. They do not provide any guarantee that participants will indeed get paid beyond the fact that the initiator will have no reason for not paying them. Furthermore, computationally expensive offline zero knowledge computations are required in the case of a message-based mixnet protocol [20], which renders the system inefficient and thus currently non-deployable.

8 Conclusions

Current anonymity networks appear to lack wide participation due to their volunteer nature. We posit that by providing economic incentives, we can help incentivize users to both participate and to use anonymity networks to protect their communications. Unfortunately, current payment schemes cannot be used to enable payments in Tor. To address this, we introduce a novel hybrid scheme and prove that it is possible to add a secure payment scheme to an onion-based anonymity network. Our approach combines features of existing payment schemes in an innovative

way, achieving provable sender-receiver unlinkability, accountability and efficiency at the same time.

Furthermore, we relate the anonymity of the overall architecture to the amount of traffic that has been forwarded through the network and the number of Tor relays. To avoid exposure, we provide initial lower-bound on the minimum payment deposit time required. Additionally – and similar to Tor – it appears that longer paths have a higher risk of including malicious nodes that may try to expose sender and receiver. On the other hand, shorter paths are more robust, incurring lower communication and computation overhead. These two limitations, namely the path length and the presence of malicious nodes, are also part of the underlying Tor network and reasonable parameters for the scheme can minimize their effect. Finally, a preliminary evaluation of our scheme indicates that PAR does not incur prohibitive communication and computational costs that could prevent its practical deployment.

Acknowledgment

We are grateful to Steven Murdoch and the referees for useful remarks and suggestions regarding this work.

References

1. Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Psuedonyms. Communications of the ACM (1981)
2. Goldschlag, D.M., Reed, M.G., Syverson, P.F.: Hiding routing information. In Anderson, R., ed.: Workshop on Information Hiding, Springer-Verlag (May 1996) 137–150 LLNCS 1174.
3. Back, A., Goldberg, I., Shostack, A.: Freedom systems 2.1 security issues and analysis. White paper, Zero Knowledge Systems, Inc. (May 2001)
4. Boucher, P., Shostack, A., Goldberg, I.: Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc. (December 2000)
5. Chaum, D.: Achieving Electronic Privacy. Scientific American (August 1992) 96–101
6. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: Proceedings of CRYPTO '88. (1988)
7. Micali, S., Rivest, R.L.: Micropayments revisited. In: CT-RSA. (2002) 149–163
8. Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the IEEE Symposium on Security and Privacy. (2006)
9. Kesdogan, D., Agrawal, D., Pham, V., Rautenbach, D.: Fundamental limits on the anonymity provided by the mix technique. In: Proceedings of the IEEE Symposium on Security and Privacy. (2006)
10. Franz, E., Jerichow, A., Wicke, G.: A payment scheme for mixes providing anonymity. In: TREC '98: Proceedings of the International IFIP/GI Working Conference on Trends in Distributed Systems for Electronic Commerce, London, UK, Springer-Verlag (1998) 94–108

11. Figueiredo, D.R., Shapiro, J.K., Towsley, D.: Using payments to promote cooperation in anonymity protocols (2003)
12. Reiter, M.K., Wang, X., Wright, M.: Building reliable mix networks with fair exchange. In: ACNS. (2005) 378–392
13. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (August 2004)
14. Tunstall, J.: Electronic currency. In: Proceedings of the IFIP WG 11.6 International Conference. (October 1989)
15. Hayes, B.: Anonymous one-time signatures and flexible untracable electronic cash. In: AusCrypt '90: A Workshop on Cryptology, Secure Communication and Computer Security. (January 1990)
16. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Advances in Cryptology - EUROCRYPT 2005. Volume 3494 of Lecture Notes in Computer Science., Springer-Verlag (2005) 302–321
17. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: TCC. (2006) 80–99
18. Dierks, T., Allen, C.: The TLS protocol version 1.0. RFC 2246 (January 1999)
19. Rivest, R.: Peppercoin micropayments. In: Proceedings of Financial Cryptography '04. Volume 3110. (February 2004) 2–8
20. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: International Workshop on Design Issues in Anonymity and Unobservability. (2001)