Distributed Double Spending Prevention*

Jaap-Henk Hoepman

TNO Information and Communication Technology
P.O. Box 1416, 9701 BK Groningen, The Netherlands
jaap-henk.hoepman@tno.nl
and
Institute for Computing and Information Sciences
Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, the Netherlands
jhh@cs.ru.nl

Abstract. We study the problem of *preventing* double spending in electronic payment schemes in a *distributed* fashion. This problem occurs, for instance, when the spending of electronic coins needs to be controlled by a large collection of nodes (e.g., in a peer-to-peer (P2P) system) instead of one central bank. Contrary to the commonly held belief that this is fundamentally impossible, we propose several solutions that do achieve a reasonable level of double spending prevention, and analyse their efficiency under varying assumptions.

1 Introduction

Many electronic payment schemes exist. For an overview, we refer to Asokan *et al.* [AJSW97] or O'Mahony *et al.* [OPT97]. Some of those are coin based, where some bitstring locally stored by a user represents a certain fixed value.

Coin based systems run the risk that many copies of the same bitstring are spent at different merchants. Therefore, these systems need to incorporate *double spending* prevention or detection techniques. To *prevent* double spending, a central bank is usually assumed which is involved in each and every transaction. In off-line scenarios (where such a connection to a central bank is not available), double spending *detection* techniques are used that will discover double spending at some later time, and that allow one to find the perpetrator of this illegal activity. A major drawback of double spending detection techniques is the risk that a dishonest user spends a single coin a million times in a short period of time before being detected. This is especially a problem if such a user cannot be punished for such behaviour afterwards, e.g., fined, penalised judicially, or being kicked from the system permanently.

* This research is/was partially supported by the research program Sentinels (www.sentinels.nl), project JASON (NIT.6677). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.
Id: double-spending.tex 17 2008-02-06 13:53:15Z jhh

Recently, the use of electronic payment like systems has been proposed[1] to counter SPAM [Hir02] or to enforce fairness among users of peer-to-peer (P2P) networks [YGM03, VCS03, GH05]. In such systems it is unreasonable to assume a central bank, either because it does not exist, or because it would go against the design philosophy of the system (as is the case for P2P networks). At first sight it then appears to be impossible to prevent double spending. This would limit the usefulness of such approaches because of the rapid double spending problem described above: users can easily rejoin a P2P system under a different alias and continue their bad practises forever.

In [GH05] we wrote:

> We note that for any system offering off-line currency, double-spending *prevention* is generally speaking not possible, unless extra assumptions (e.g., special tamper proof hardware) are made.

In that paper, in fact, we were not considering a completely off-line system, but a decentralised system without a central bank instead. The difference turns out to be decisive. In a truly off-line system (where the receiver of a coin has no network access to perform any kind of checking, and where the spender of a coin is not forced to adhere to a security policy through some kind of tamper proof hardware [SS99]) the chances of double spending prevention are slim. We soon after realised, however, that the situation is not so bad in an on-line but decentralised system without a central bank.

The crucial observation is that it may be impossible, or very expensive, to prevent every possible double spending of a coin (i.e., a deterministic approach), but that it may very well be possible to prevent that a particular coin is double spent *many times*, using efficient randomised techniques. Even such a weaker guarantee limits the damage an adversary can do. In other words, the main paradigm shift is the realisation that double spending a single coin twice is not so bad, but spending it a hundred times should be impossible. Of course, such a probabilistic and limited security property may not be strong enough for the protection of 'real' money. It may, however, be quite workable for currencies used to enforce fairness among P2P users.

In this paper we study several such techniques for distributed double spending prevention. We focus in this paper on methods to distribute the tasks of the central bank over (a subset of) the nodes in the system. An extreme case would be the distribution of the central bank over all nodes in the system, making everyone a clerk working for the bank. This would lead to an enormous communication overhead, as all n nodes in the system would have to be contacted for each and every transaction. We study techniques to reduce the size of such clerk sets, mainly in probabilistic ways, while still keeping reasonable double-spending prevention guarantees.

[1] America Online and Yahoo announce introduction of electronic postage for email messages ("Postage is Due for Companies Sending E-Mail", New York Times, February 5, 2006).

Next to a deterministic approach, there are two fundamentally different ways to construct the clerk sets in a probabilistic manner. The most efficient method — yielding the smallest clerk sets — uses the unqiue identifier of a coin to limit the possible members of the clerk set in advance. In this model, certain clerks attract certain coins, making it far more likely that double spending is detected. The drawback is that given a particular coin these clerks are known beforehand. This means the adversary has advance knowledge regarding the clerks that it needs to bribe in order to be able to double spend a particular coin. In certain situations this may be undesirable. Therefore we also study the less efficient case where the clerks are selected uniformly at random.

1.1 Our results

We prove the following results, where n is the total number of nodes, f is the total number of dishonest nodes, d is the number of dishonest nodes that may be corrupted by the adversary after they join the network, and s is the security parameter (see Section 2 for details).

Deterministic double spending prevention can be achieved with clerk sets of size $2\sqrt{n(f+1)}$.

Using randomisation double spending can be prevented with clerk sets of size at least $\sqrt{\frac{ns}{\log e(1-f/n)}}$. If we require that double spending only needs to be detected when a single coin is double spent at least r times[2] we need clerk sets of size at least $\frac{\sqrt{2ns}}{r}$ when $f=1$ (i.e., if only the double-spender itself is dishonest) and $\sqrt{\frac{ns}{\log e(1-f/n)r}}$, when $f>1$. Note that it is indeed interesting to consider the $f=1$ case seperately, because it corresponds to the situation where nodes in the clerk sets have no incentive to collaborate with the double spender to let him get away undetected, and is closely related to the selfish but rational models used in game theoretic analysis of security protocols (cf. [IML05]).

Finally we prove that making use of the coin identifier to construct coin specific clerk spaces of size β at least $d+\frac{s}{\log((n-d)/(f-d))}$ clerk sets sampled from this space of size at least $\frac{\beta}{r\log e}(s+1+\log(r+2))$ suffice to detect a coin that is double spent at least r times.

These results tell us the following. Deterministically, clerk sets that have $\sqrt{nf}$ nodes suffice. For any reasonable f this is unworkable. Using randomisation, $\sqrt{n/(1-f/n)}$ is good enough. For decent fractions of faulty nodes (e.g., $f/n=1/2$) this stays $O(\sqrt{n})$. When we relax the double spending detection requirement and allow upto r double spendings to be undetected, clerk sets can be further reduced by a $\sqrt{r}$ factor. Finally, if we use information stored in the coin, the size of the clerk sets becomes independent of the size of the network, depending only on the inverse ratio n/f of faulty nodes, and the number of corruptable nodes d.

[2] r denotes the number of times a coin is double spent. To be precise, when a node spends the same coin x times, then $r=x-1$.

1.2 Related research

The deterministic variant of distributed double spending prevention, i.e., the one where double spending is *always* prevented, is equivalent to the problem of distributing a database over n nodes, f of which may be faulty. Quorum systems (cf. [MR98, MRWW01]) have been studied as an abstraction of this problem, to increasing the availability and efficiency of replicated data. A quorum system is a set of subsets (called quorums) of servers such that every two subsets intersect. This intersection property guarantees that if a write-operation is performed at one quorum, and later a read-operation is performed at another quorum, then there is some server that observes both operations and therefore is able to provide the up-to-date value to the reader. The clerk sets in our work correspond to the quorums in that line of research. We do note however that the relaxation of allowing upto r double spendings to occur is not covered by the work on quorum systems.

Our approach is in a sense a dual to the one advocated by Jarecki and Odlyzko [JO97] (and similarly by Yacobi [Yac99]), in which double spending is prevented probabilistically and efficiently by checking a payment with the *central* bank only with some probability (instead of always).

1.3 Structure of the paper

The paper is organised as follows. We first describe the model and the basic system architecture in Section 2. This fixes the way coins are represented and spent among nodes, and describes how clerk sets are used to detect double spending. This architecture is independent of how the clerk sets are constructed. Different construction methods yield different performance, as described in the sections following. It is exactly these combinatorial constructions that are the main contributions of this paper.

We analyse the performance of fixed clerk sets in Section 3, followed by the analysis of randomly chosen clerk sets in Section 4. Next, in Section 5, we study what happens if we allow coins to be double spend more often, up to a certain limit r. Then, in section 6 we discuss ways to further reduce the size of the clerk sets by making use of information in the coin. We conclude with a thorough discussion of our results in Sect. 7.

2 Model and notation

We assume a distributed system consisting of n nodes, at most f of which are dishonest. The dishonest nodes are under the control of the adversary. If the system is a peer-to-peer (P2P) overlay network, the nodes receive a random identifier when joining. This identifier is not under the control of the adversary. The adversary may, however, be able to compromise d out of the f dishonest nodes *after* joining the network, i.e., it may compromise at most d nodes for which it knows the P2P identifier[3].

Each node owns a pair of public and private keys. A signature $[m]_i$ of node i on a message m can be verified by all other nodes. We let log denote the logarithm base 2.

The system handles coins, that are uniquely identified by a coin identifier *cid*. Valid coin identifiers cannot 'easily' be generated by nodes themselves. Nodes can distinguish valid coins from invalid ones. A detailed discussion on how nodes initially obtain such coins lies outside the scope of this paper. But to argue the viability of our approach, we briefly mention the following two options. Coins could, for instance, be distributed initially by a central authority. In this case, the coin identifier incorporates a digital signature from this authority. Or they could be generated by the nodes themselves by finding collisions in a hash function h (cf. [GH05]). Then, the coin identifier contains the pair x, y such that $h(x) = h(y)$.

Nodes communicate by exchanging messages. We assume a completely connected network, or a suitable routing overlay. The network is asynchronous. In particular, coins may be spent concurrently. The network is static: no nodes join or leave the network once the system runs.

All in all these are quite strong assumptions (a static network, with a network wide PKI, and a point-to-point communication substrate), but not unreasonably so. In any case, they allow us to focus on the main research issue: the combinatorial analysis of distributing the task of an otherwise centralised bank over the nodes of a distributed system, such that double spending is prevented.

The adversary tries to double spend a single coin at least r times (when a node spends a single coin x times, then $r = x - 1$). We say the system is secure with security parameter s if the adversary must perform an expected $O(2^s)$ amount of work in order to be successful. We show this by proving that the probability of success for the adversary for a single try is at most 2^{-s}.

We note that we do not consider denial of service attacks, for example attacks where the clerk sets receive polluted information from dishonest nodes to invalidate coins held by honest nodes.

2.1 Distributing the bank

Throughout the paper we assume the following system architecture to distribute the bank over the nodes in the network.

A coin is uniquely determined by its coin-id *cid*. Spending a coin c_i transfers ownership of that coin from a sender s to a receiver r. We use the following method (also depicted in Figure 1): the receiver sends a nonce z to the sender, who then signs the coin, together with the nonce and the name of the receiver, sending the result

$$c_{i+1} = [c_i, z, r]_s$$

[3] This distinction between f and d turns out to be only significant in the case where coin identifiers are used to restrict the size of the clerk sets.

sender s		**receiver** r		**clerk** b
	$\xleftarrow{z}$	Generate nonce z		
Spend coin c				
Delete coin c	$\xrightarrow{[c_i,z,r]_s}$	Receive c_{i+1}. Verify nonce and signature. Obtain $B_{r,c_{i+1}}$ $C = \varnothing$ Foreach $b \in B_{r,c_{i+1}}$:	$\xrightarrow{c_{i+1}}$	
				Lookup $cid(c_{i+1})$ in DB_b. Insert c_{i+1} in DB_b. $C' =$
		add C' to C. when all C' received Accept if $c \Rightarrow c_{i+1}$ forall $c \in C$.	$\xleftarrow{C'}$	$\{c \in DB_b \mid cid(c) = cid(c_{i+1})\}$

Fig. 1. Coin spending and detection protocol.

back to the receiver. We call c_i the immediate prefix of c_{i+1} (denoted $c_i \rightarrow c_{i+1}$), and require that s equals the receiver of c_i (otherwise c_i should not have been in the posession of s in the first place). An unspent coin simply corresponds to its coin-id *cid*. c is a prefix of c', denoted $c \Rightarrow c'$ if there is a sequence of coins $c_0, \ldots, c_k$, $k > 0$ such that $c = c_0$, $c_k = c'$ and $c_i \rightarrow c_{i+1}$ for all $0 \leq i < k$. The coin-id $cid(c)$ of a coin equals its shortest prefix, or c itself if no prefix exists.

So called *clerk sets* are used to verify the validity of a coin. These clerk sets consist of nodes in the network that simulate a bank in a distributed fashion. The selection of nodes that are member of a clerk set $B_{r,c}$ can be either done deterministically or randomly, and may depend on both the node r accepting the coin and the coin identifier $cid(c)$ of the coin being accepted. To perform their duties, the nodes in a clerk set store the history of coins. When a receiver r receives a coin c, it first verifies the signature, the nonce, and the sender. It then requests from each clerk in the clerk set $B_{r,c}$ all coins with coin-id $cid(c)$ that it stores. At the same time, the clerks store c. These two steps are one atomic operation. If all coins r receives from its clerk set are proper prefixes of c, it accepts the coin. Otherwise it rejects the coin.

We note that the size of a coin increases every time it is spent, because of the signature that must be added. Similarly, the set of coins stored by the clerk sets grows without bounds. Dealing with these unbounded space requirements falls outside the scope of this paper. We discuss some ways to bound the space requirements in Sect. 7.

The remainder of this paper assumes the above protocol for spending a coin, and is merely concerned with different methods for obtaining $B_{r,c_{i+1}}$ such that double spending is prevented. The following property of the system described above is the basis for the main results of this paper.

Property 2.1. Let j and k be honest nodes, and let c be a coin. If $B_{j,c} \cap B_{k,c}$ contains at least one honest node, then no node can double spend a coin with coin-id $cid(c)$ at both j and k using the protocol described above.

Proof. Let x be the honest node in $B_{j,c} \cap B_{k,c}$. If i manages to double spend c at both j and k ($j = k$ is possible), x receives a request to lookup (and immediately store) $c_j = [c', z_j, j]_i$ from j and $c_k = [c'', z_k, k]_i$ from k (with unique nonces z_j and z_k) where $cid(c_j) = cid(c_k)$, $c_j \not\rightsquigarrow c_k$ and $c_k \not\rightsquigarrow c_j$ (by definition of double spending). W.l.o.g. assume j makes that request to x first. Then j stores c_j at DB_x before k requests all coins with $cid(c) = cid(c_k)$. Then k retrieves c_j with $c_j \not\rightsquigarrow c_k$ and hence k does not accept c_k. □

Observe that the inclusion of nonces in the coin spending phase is really only necessary to determine the exact node that double-spent the coin first.

3 Fixed clerk sets: deterministic case

We will now study several methods to assign clerk sets to nodes. We start with the deterministic case where each node is given a fixed clerk set B_i. We assume $d = f$ (in the deterministic case it makes no difference whether the adversary can corrupt the nodes after they join the network or only before that: it can ensure *in advance* to only double spend at nodes for which the clerk sets contain no honest nodes).

If, except for the node trying to double spend, there are no dishonest nodes, we only need to require $B_i \cap B_j \neq \varnothing$ (and the double spender should not be the only node in that intersection). Clearly, we can set $B_i = \{b\}$ for all i and some clerk b. This coincides with the 'central bank' case described in the introduction. In this paper we are of course interested in the distributed case, where there should be no single point of failure, and where the load for preventing double spending is evenly distributed over *all* participating nodes. The optimal construction of such sets was already studied in the context of the *distributed match-making* problem by Mullender and Vitányi [MV88, EFF85]. They show that an assignment of sets exists such that $|B_i| \leq 2\sqrt{n}$ for all i, while for all i, j $B_i \cap B_j \neq \varnothing$. They also prove a matching lower bound[4].

Now suppose we do have f dishonest nodes. Using the techniques outlined above, we arrive at the following bound.

[4] Note that if we somehow could construct a 'uniform, randomised' selection of the node responsible for keeping track of the current owner of a coin, then using this single node as the clerk set for that coin would implement a distribution solution to the problem. This is studied in more detail in section 6.

Theorem 3.1. *Double spending is deterministically prevented with fixed clerk sets of size $2\sqrt{n(f+1)}$, when there are at most f dishonest nodes.*

Proof. To guarantee detection of double spending we need at least $f+1$ clerks in the intersection of any two clerk sets, hence

$$|B_i \cap B_j| > f \ .$$

One way to approach this extension is as follows. Cluster the n nodes into groups of $f+1$ nodes each (for simplicity assume $f+1$ exactly divides n). For the resulting $\frac{n}{f+1}$ so-called supernodes N_i, create super clerk sets $\mathbf{B}_i$ as before. Now for each original node i, let its clerk set be the union of the nodes in the super nodes that are a member of its super clerk set $\mathbf{B}_i$. In other words, let j be a member of super node N_i. Then

$$B_j = \bigcup_{N_k \in \mathbf{B}_i} N_k \ .$$

We know $|\mathbf{B}_i| = 2\sqrt{\frac{n}{f+1}}$, and that each super node covers $f+1$ nodes. Hence $|B_j| \leq 2\sqrt{n(f+1)}$. By construction, for any pair i, j there is an $N_k \in B_i \cap B_j$. Hence $|B_i \cap B_j| > f$. □

4 Random clerk sets

We now consider the case where each time a node i receives a coin it generates a different random clerk set B_i to verify that the coin is not being double spent[5]. Now suppose we have f dishonest nodes. Again we assume $d = f$ (because the clerk sets are regenerated every time a coin is received, the adversary gains no advantage if it is able to corrupt some nodes right after system initialisation).

Theorem 4.1. *Double spending is prevented with overwhelming probability using random clerk sets of size at least $\sqrt{\frac{ns}{\log e(1-f/n)}}$.*

Proof. Let B_i be given, and randomly construct B_j. Let b be the size of the clerk sets that we aim to bound. B_j does not prevent double spending if it only contains nodes not in B_i, unless they are dishonest. To simplify analysis, let us assume that in the random construction of the set B_j (and the given set B_i) we are sampling with replacement. This way we overestimate the probability of constructing such a bad set (because we do not reduce the possible number of bad choices that would occur with sampling *without* replacement). We will then show that even with this overestimation, this event will occur with probability at most 2^{-s}.

[5] Actually, in this case a node can use the same randomly generated clerk set throughout, *provided* that $d = 0$. This is no longer the case when we allow small multiple spendings, analysed in Section 5.

For each member x of B_j, we should either pick a node not in B_i (with probability $\frac{n-b}{n}$), or if we do (with probability $\frac{b}{n}$), this node should be dishonest. Each node in B_i has probability $\frac{f}{n}$ to be dishonest. Hence

$$\mathbf{Pr}\left[x \text{ is bad}\right] = \frac{n-b}{n} + \frac{b}{n}\frac{f}{n} .$$

Then

$$\mathbf{Pr}\left[B_j \text{ is bad}\right] = \left(\mathbf{Pr}\left[x \text{ is bad}\right]\right)^b = \left(\frac{n-(1-f/n)b}{n}\right)^b .$$

With $(1-\frac{1}{x})^x < e^{-1}$, the latter can be bounded from above by $e^{-\frac{1-f/n}{n}b^2}$. We require $\mathbf{Pr}\left[B_j \text{ is bad}\right] \leq 2^{-s}$. This is achieved when

$$e^{-\frac{1-f/n}{n}b^2} < 2^{-s} .$$

Taking logarithms and rearranging proves the theorem. □

This improves the deterministic case, where we have a $\sqrt{f}$ dependence on f.

5 When coins get spent more often

Clearly, the problem of double spending becomes more pressing when coins are double spent (much) more than once. We will now show that this can be prevented with high probability with even small clerk sets. Note that multiple double spending only helps reducing the size of the clerk sets in the randomised case: in the deterministic case either the first double spending is prevented straight away, or no double spending is prevented at all.

Let r be the number of times a single coin is double spent by the same node[6] We first consider the failure free case, i.e., except for the node trying to double spend, there are no dishonest nodes. This case captures the situation where nodes in the clerk sets have no incentive to collaborate with the double spender to let him get away undetected, and is closely related to the selfish but rational models used in game theoretic analysis of security protocols (cf. [IML05]).

Theorem 5.1. *When only the owner of a coin is dishonset, double spending of a single coin at least r times is prevented with overwhelming probability using random clerk sets of size b such that $b > \frac{\sqrt{2ns}}{r} + 1$ (or $b > \frac{n-1}{r+1}$).*

Proof. Let B_i be the set used for the verification of the coin when it is spent for the i-th time. Let q be the node double spending. There are $r+1$ such sets if the coin is double spent r times. If double spending is not detected one of those r times, the adversary wins. This happens when $B_i \cap B_j$ contains at most the double spender q itself, for all pairs i, j. The probability that this happens is

[6] Recall that when a node spends the same coin x times, then $r = x - 1$.

computed as follows (where we assume $(r+1)b \leq n$ or else such a collection of sets simply does not exist).

After constructing the i-th set such that none of the i sets (each with b members) do mutually intersect except on the double spender q, there are at most $n-i(b-1)$ nodes to choose from for the $i+1$-th set, and the probability that this set does not intersect the i others except on q becomes at most $\binom{n-i(b-1)}{b}/\binom{n}{b}$. Expanding binomials to their factorial representation, and cancelling factorials in nominators and denominators, we conclude that this is less than

$$\left(\frac{n-i(b-1)}{n-b+1}\right)^b \ .$$

Hence

$$\mathbf{Pr}\left[\text{double spending not detected}\right] \leq \prod_{i=1}^{r} \frac{\binom{n-i(b-1)}{b}}{\binom{n}{b}} \leq \prod_{i=1}^{r} \left(\frac{n-i(b-1)}{n-b+1}\right)^b \ .$$

Further simplification using $\frac{a-b}{n}\frac{a+b}{n} \leq \frac{a^2}{n^2}$ shows that this is bounded from above by

$$\left(\frac{n-\frac{r+1}{2}(b-1)}{n-b+1}\right)^{rb} \ .$$

We want this latter expression to be negligible, i.e., less than 2^{-s}. Inverting fractions and taking logarithms this leads to the inequality

$$rb\log\left(\frac{n-b+1}{n-\frac{r+1}{2}(b-1)}\right) > s \ .$$

Using $(r+1)b \leq n$ we see $\frac{n-b+1}{n-\frac{r+1}{2}(b-1)} \leq 2$. Using this, and the fact that $\log(1+x) \geq x$ for all x between 0 and 1, we have

$$\log\left(\frac{n-b+1}{n-\frac{r+1}{2}(b-1)}\right) \geq \left(\frac{\frac{r-1}{2}(b-1)}{n-\frac{r+1}{2}b}\right)$$

Hence we require

$$rb\left(\frac{\frac{r-1}{2}(b-1)}{n-\frac{r+1}{2}b}\right) > s$$

Simplifying this proves the theorem. □

Next, we consider the case when there are at most $f > 1$ dishonest nodes.

Theorem 5.2. *Double spending of a single coin at least r times is prevented with overwhelming probability using random clerk sets of size at least* $\sqrt{\frac{ns}{\log e(1-f/n)r}}$.

Proof. Again, let there be $r+1$ sets B_i, each used for the verification of the coin when it is spent for the i-th time. Let F denote the set of faulty nodes. If double spending is not detected one of those $r+1$ times, the adversary wins. This happens when

$$(B_i \cap B_j) \setminus F = \varnothing, \text{for all } i, j \ .$$

We are going to estimate the probability that this happens by only considering $B_1 \cap B_j \setminus F = \varnothing$ for all $j \neq 1$. Then

$$\begin{aligned}\mathbf{Pr}\left[\text{double spending not detected}\right] &< \left(\mathbf{Pr}\left[B_1 \cap B_j \setminus F = \varnothing\right]\right)^r \\ &< \left(\mathbf{Pr}\left[x \notin B_1 \vee x \in F\right]^b\right)^r \ ,\end{aligned}$$

where in the last step we consider arbitrary x and sample with replacement. This latter probability is, like the proof in Theorem 4.1

$$\mathbf{Pr}\left[x \text{ is bad}\right] = \frac{n-b}{n} + \frac{b}{n}\frac{f}{n} \ .$$

Proceeding similar to that proof, we obtain $b > \sqrt{\frac{ns}{\log e(1-f/n)r}}$. □

The bound appears not to be tight (in fact it is worse than Theorem 5.1 by a factor $\sqrt{r}$) because we only estimated the probability that no clerk set intersects with the first clerk set, thus greatly exaggerating the success of the adversary. Simulations suggest that the size of the clerk sets b is indeed inversely proportional to the number of clerk sets r even when faulty nodes exist.

6 Coin-specific clerk sets

Up till now, we have assumed that clerk sets are constructed independent of the coin that needs to be checked. This is a restriction. In fact, we will now show that under certain circumstances, the use of the coin identifier in the construction of the clerk sets may help reducing the size of the clerk sets even further.

In previous work on digital karma [GH05] we investigated the design of a decentralised currency for P2P networks with double-spending *detection*. We showed the following result, given an assignment of β nodes derived from a coin identifier *cid* by

$$\mathbf{B}_{cid} = \{h^i(cid) \bmod n \mid 1 \leq i \leq \beta\}$$

(where we ignore the possibility of collisions for the moment) where h is a random hash function.

Lemma 6.1 ([GH05]). *If $\beta > d + \frac{s}{\log((n-d)/(f-d))}$, then $\mathbf{B}_{cid}$ contains only dishonest nodes with probability less than 2^{-s}.*

Note that in the proof of this result we use the fact that the adversary controls at most d nodes for which it knows membership of a particular set $\mathbf{B}_{cid}$; for all other $f-d$ dishonest nodes membership of this set is entirely random.

Using this new approach as a starting point, we now analyse how frequent double spending of a single coin can be prevented more efficiently.

Clearly, when there are no dishonest nodes, the single node clerk set $B_{cid} = \{h(cid)\}$ suffices to prevent double spending (provided of course that the coin is never spent by this particular node itself). This is a distributed solution because the hash function distributes the clerk assignment uniformly over all available nodes.

Similarly, using the Lemma 6.1, we see that using $\mathbf{B}_{cid}$ as the clerk set each time coin *cid* is spent, double spending is prevented with overwhelming probability as well, even if the adversary gets to corrupt d out of f nodes of his own choosing. This is summarised in the following theorem.

Theorem 6.2. *Double spending is prevented with overwhelming probability using clerk sets derived from a coin identifier, of size at least* $\beta > d + \frac{s}{\log((n-d)/(f-d))}$.

But we can do even better than that if we are willing to allow a coin to be double spent at most r times. The idea is to start with the coin-specific clerk space $\mathbf{B}_{cid}$ of size β, but to use a smaller random subset $B_i \subset \mathbf{B}_{cid}$ of size b as the clerk set to use when spending the coin for the i-th time.

Observe that the size of the clerk space now is more or less independent of n: it only depends on the fraction of dishonest nodes. Compared to the original randomised clerk set case (see Theorem 4.1) when setting $d = 0$ we see that β increases much less rapidly with increasing fraction of dishonest nodes. Note that reducing the sample space in this original case from n to say n' would improve the bound; however, the solution would no longer be distributed because certain nodes *never* would become members of a clerk set.

Theorem 6.3. *Double spending of a single coin cid at least r times is prevented with overwhelming probability using coin specific clerk spaces of size β at least $d + \frac{s}{\log((n-d)/(f-d))}$ and clerk sets of size b at least* $\frac{\beta}{r \log e}(s + 1 + \log(r + 2))$

Proof. Consider an arbitrary coin with coin identifier *cid*. Let $\beta = |\mathbf{B}_{cid}|$. From Lemma 6.1 we know that if $\beta > d + \frac{s+1}{\log((n-d)/(f-d))}$, then $\mathbf{B}_{cid}$ contains no honest nodes with negligible probability $2^{-(s+1)}$.

Let this coin be double spent $r > 1$ times, and let $B_i \subset \mathbf{B}_{cid}$ be a random subset of size b that serves as the clerk set to use when spending the coin for the i-th time. We will show that when $\mathbf{B}_{cid}$ contains at least one honest node x, the probability that x is not a member of at least two sets B_i and B_j is again at most $2^{-(s+1)}$. Multiplying these two probabilities we can conclude that the adversary can only succeed spending the coin r times with probability at most 2^{-s}, which proves the theorem.

We bound the probability that x is not a member of at least two sets B_i and B_j as follows. We have

$$\mathbf{Pr}\left[x \notin B_i\right] = \frac{\beta-1}{\beta}\frac{\beta-2}{\beta-1}\cdots\frac{\beta-b}{\beta-b+1} = 1 - \frac{b}{\beta}\ .$$

Call this probability p. Then $q = 1 - p = \frac{b}{\beta}$. Let X be a random variable denoting the number of sets B_i of which x is a member. Then

$$\mathbf{Pr}[X \leq 1] = p^{r+1} + \binom{r+1}{1} p^r q \ .$$

Assume for the moment that $b > \beta/2$. Then $q > p$ and hence $\mathbf{Pr}[X \leq 1] \leq (r+2)qp^r$, which should be less than $2^{-(s+1)}$. Substituting the values for p and q and using $\frac{b}{\beta} \leq 1$, this is achieved when

$$(r+2)\left(1 - \frac{b}{\beta}\right)^r \leq 2^{-(s+1)} \ .$$

Using $(1 - 1/x)^x \leq 1/e$ and taking logarithms we need

$$\log(r+2) - r \log e \frac{b}{\beta} \leq -(s+1)$$

From this the theorem follows. □

The proof of this theorem uses a rather crude approximation of the probability that an adversary can cheat. In fact, it is far more likely that a coin specific clerk space contain more than one honest node, making it harder for the adversary to avoid them in the r clerk sets.

7 Conclusions & Further Research

Interestingly, the probability of polling the central bank in the scheme of Jarecki and Odlyzko [JO97] is proportional to the amount of the transfer, such that the number of polling messages is constant for a given amount of credit: whether a user spends all her credit in a few big transactions, or many micro payments does not matter. To get a similar property in our scheme would require us to change the size of the clerk sets depending on the amount of the transaction (i.e., the value of the coin, if there are multi valued coins in the system), or to contact the clerk sets only with a certain probability for each transaction. Further research is necessary to explore these ideas and to determine their impact on the efficiency of double spending prevention in a decentralised, distributed currency scheme.

The current analysis is based on a few strong assumptions. For one thing, we assume that the network is static. To fully apply our ideas to for instance P2P networks requires us to take dynamic node joins and leaves into account. Also, we assume transmitting coins is an atomic operation. Probably, the coin transfer protocol becomes slightly more involved when we need to handle concurrent coin spending. Finally, the coin transfer protocol assumes that coins can grow unbounded in size: with every transfer of a coin, it gains another signature. Methods to reduce the space complexity should be investigated. This is not easy however, because the double spending prevention system depends on a more or less correct notion of time, and aims to record who owns which coin at what time. Preventing nodes to warp the coins they own into the future (and thus bypassing all double spending prevention) is not trivial. We do note however, that clerks only need to store the coin with the longest prefix for a particular coin identifier.

Finally, there are other interesting approaches that might be useful to implement distributed double spending prevention.

One approach is to try to limit the rate at which nodes can spend coins in the first place. HashCash [Bac97] could be used to do this. In this setting, a node wishing to spend a coin is forced to spend a non-negligible amount of work first to compute some function, e.g., by finding a collision in a moderately strong hashfunction. The receiver of the coin verifies the function result and only accepts the coin when the result is correct. If a lower bound on the actual time needed to compute the function is known (and this is not always easy given the diversity of hardware platforms), this implies an upper bound on the amount of money a coin spent (and therefore double spend).

References

[AJSW97] Asokan, N., Janson, P. A., Steiner, M., and Waidner, M. The state of the art in electronic payment systems. *IEEE Computer* **30**, 9 (1997), 28–35.

[Bac97] Back, A. Hashcash - a denial of service counter-measure. http://www.cypherspace.org/hashcash, 1997.

[EFF85] Erdös, P., Frankl, P., and Füredi, Z. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics* **51**, 1–2 (1985), 79–89.

[GH05] Garcia, F. D., and Hoepman, J.-H. Off-line karma: A decentralized currency for peer-to-peer and grid networks. In *3rd ACNS* (New York, NY, USA, 2005), J. Ioannidis, A. Keromytis, and M. Yung (Eds.), LNCS 3531, Springer, pp. 364–377.

[Hir02] Hird, S. Technical Solutions for Controlling Spam. In *AUUG2002* (Melbourne, 2002).

[IML05] Izmalkov, S., Micali, S., and Lepinski, M. Rational secure computation and ideal mechanism design. In *46th FOCS* (2005), IEEE Comp. Soc. Press, pp. 585–595.

[JO97] Jarecki, S., and Odlyzko, A. An efficient micropayment system based on probabilistic polling. In *1st Int. Conf. Fin. Crypt.* (Anguilla, British West Indies, 1997), R. Hirschfeld (Ed.), LNCS 1318, Springer, pp. 173–191.

[MR98] Malkhi, D., and Reiter, M. Byzantine quorum systems. *Distributed Computing* **11**, 4 (1998), 203–213.

[MRWW01] Malkhi, D., Reiter, M., Wool, A., and Wright, R. Probabilistic Quorum Systems. *Information and Computation* **170**, 2 (2001), 184–206.

[MV88] Mullender, S. J., and Vitányi, P. M. B. Distributed match-making. *Algorithmica* **3** (1988), 367–391.

[OPT97] O'Mahony, D., Peirce, M., and Tewari, H. *Electronic Payment Systems*. Artech House, 1997.

[SS99] Schneier, B., and Shostack, A. Breaking up is hard to do: Modelling security threats for smart cards. In *1st USENIX Worksh. on Smartcard Tech.* (Chicago, IL, 1999), USENIX, pp. 175–185.

[VCS03] Vishnumurthy, V., Chandrakumar, S., and Sirer, E. G. KARMA: a secure economic framework for peer-to-peer resource sharing. In *Proc. Workshop on the Economics of Peer-to-Peer Systems* (Berkeley, California, 2003). Papers published on http://www.sims.berkeley.edu/research/conferences/p2pecon/index.html.

[Yac99] Yacobi, Y. Risk management for e-cash systems with partial real-time audit. In *3rd Int. Conf. Fin. Crypt.* (Anguilla, British West Indies, 1999), M. K. Franklin (Ed.), LNCS 1648, Springer, pp. 62–71.

[YGM03] Yang, B., and Garcia-Molina, H. PPay: micropayments for peer-to-peer systems. In *10th CCS* (Washington D.C., USA, 2003), V. Atluri and P. Liu (Eds.), ACM, pp. 300–310.