Incentive-Compatibility in a Distributed Autonomous Currency System

Kenji Saito[1], Eiichi Morino[2], and Jun Murai[3]

[1] Graduate School of Media and Governance, Keio University
[2] Gesell Research Society Japan
[3] Faculty of Environmental Information, Keio University

Abstract. *Peer-to-peer complementary currencies* can be powerful tools for promoting exchanges and building sustainable relationships among selfish peers on the Internet.
i-WAT[10] is a proposed such currency based on the WAT System, a polycentric, real-life complementary currency using *WAT tickets* as its media of exchange. Participants spontaneously issue and circulate the tickets as needed, whose values are backed up by chains of trust. *i*-WAT implements the tickets electronically by exchanging messages signed in OpenPGP[3].
This paper claims that the design of *i*-WAT is incentive-compatible as to protection against moral hazards, or threats caused by selfish peers because they may take advantage of the rules; such hazards are defused in *i*-WAT if the participants react against misbehaviors of others by pursuing their own benefits.
A reference implementation of *i*-WAT has been developed in the form of an XMPP (Extensible Messaging and Presence Protocol)[4][5] instant messaging client. We have been putting the currency system into practical use since June 2004.

1 Introduction

1.1 Peer-to-Peer Complementary Currency

Exchanging is a necessary building block of peer-to-peer (P2P) systems, which can potentially harness the under-utilized power of the network of computers connected one another via the Internet. Since the resources are distributed over autonomous entities, such exchanging needs to be performed in an *incentive-compatible*[6] way: the coordination must be accomplished by collection of selfish behaviors. A medium of exchange which represents a guaranteed value should take an important role in the design of P2P systems.

Money is a well-known medium of exchange, but its scarcity has caused a lot of problems. *Complementary currencies*, or alternative forms of monetary media, have been proposed and tested in real life to achieve an autonomous, sustainable local economy even in short of money. There have been successful cases, such as experiments in Wörgl in 1932 (stamp money[15]), in Comox Valley

in 1983 (Local Exchange Trading System[16]) and in Ithaca since 1991 (Ithaca HOURs[7]).

Those complementary currencies are used to support values which are not readily circulated in today's economy, such as volunteer works or skills that are not regularly utilized. Translating them onto the Internet would benefit the design of P2P systems, which are also intended to make use of under-utilized resources. But then, those currencies also need to be peer-to-peer.

We proposed *i*-WAT[10] in year 2003 as such a currency usable on the Internet, based on the WAT System[17]. The WAT System is a system of polycentric complementary currencies using *WAT tickets* as its media of exchange. A WAT ticket is like a bill of exchange, but without a specified redemption date or place. *i*-WAT implements the tickets electronically by exchanging messages signed in OpenPGP[3]. It has been put into practical use since June 2004.

1.2 Contributions of This Paper

This paper begins by describing the core designs of WAT/*i*-WAT and the trust and incentive models of *i*-WAT. It then shows, by a game-theoretical analysis, that the design of *i*-WAT is incentive-compatible as to protection against moral hazards: taking advantage of the rules will result in the subject's confrontation to an uncontrollable risk. Since *i*-WAT has no fixed authority, such risks are imposed by rational behaviors of other participants.

The hazards in concern will include impostors, unintentional breach of trust and collusions.

2 WAT/*i*-WAT Currency System

2.1 The WAT System

Overview The WAT System[17] is a complementary currency designed by Eiichi Morino, a coauthor of this paper. It has been used broadly, especially in Japan, since its introduction in August 2000.

A *WAT ticket*, a physical sheet of paper resembling a bill of exchange, is used as the medium of exchange in the system. A lifecycle of a WAT ticket involves three stages of trading as illustrated in Fig. 1:

1. Issuing – the birth of a WAT ticket
   A *drawer* issues a WAT ticket by writing on an empty form the name of the provider (*lender*) of the goods or service, the amount of debt[1], the present date, and the drawer's signature. The drawer gives the ticket to the lender, and in return obtains some goods or service.

[1] Typically in the unit kWh, which represents cost of producing electricity from natural energy sources.

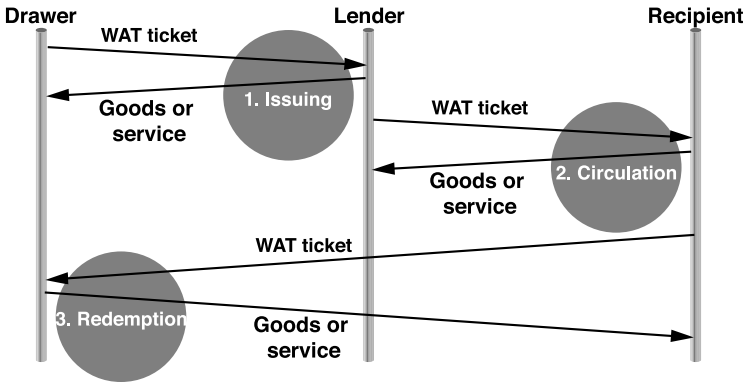


Fig. 1. Three stages of trading with a WAT ticket

2. Circulation – ordinary exchange
   The person to whom the WAT ticket was given can become a *user*, and use it for another trading. To do so, the user writes the name of the recipient, as well as their own, on the reverse side of the ticket. The recipient will become a new user, repeating which the WAT ticket circulates among people.
3. Redemption – the return of the WAT ticket
   The WAT ticket is invalidated when it returns, as a result of a trade, to the drawer.

Distinctive Features of the WAT System

Autonomy Anyone can spontaneously become a member of the WAT System with a sheet of paper if they follow the above protocol.

Compatibility A WAT ticket is compatible with any other WAT tickets in the world, so that the currency system is operable globally, as long as the drawer can be credited.

Extensibility The protocol illustrated in Fig. 1 defines *the WAT Core*, the essence of the WAT System. An *extended part* can be defined for a new currency based on the WAT System, stating, for example, the region, group and duration in which the tickets are usable, as well as the unit in which the debt is quantified.

Security In case the drawer fails to meet their promise on the ticket, the lender assumes the responsibility for the debt. If the lender fails, the next user takes over. The responsibility follows the chain of endorsements. The longer the chain is, the more firmly backed up the ticket is. Therefore the length of the chain of endorsements represents the extent of trust the ticket has gained.

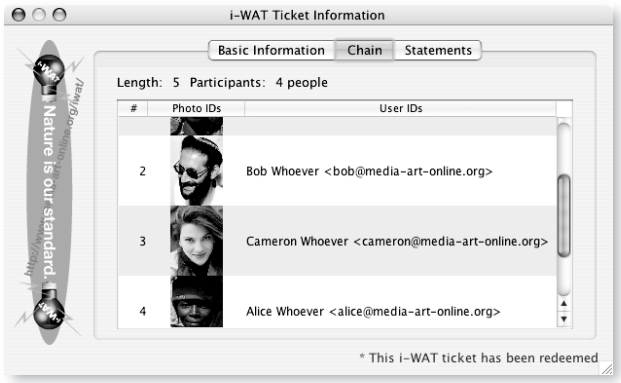


Fig. 2. Visual representation of an *i*-WAT ticket

Table 1. *i*-WAT messages

Message	Sender	Receiver	Function
<draw/>	drawer	recipient (lender)	draws an *i*-WAT ticket.
<use/>	user	recipient	uses an *i*-WAT ticket.
<accept/>	recipient	drawer and user	confirms readiness to accept the *i*-WAT ticket once it is validated.
<reject/>	recipient	drawer or user*	rejects an *i*-WAT ticket.
<approve/>	drawer	user and recipient	validates an *i*-WAT ticket, and approves the transaction.
<disapprove/>	drawer	user and recipient	denies an *i*-WAT transaction.

* depending on whether the ticket has just been issued or in circulation, respectively.

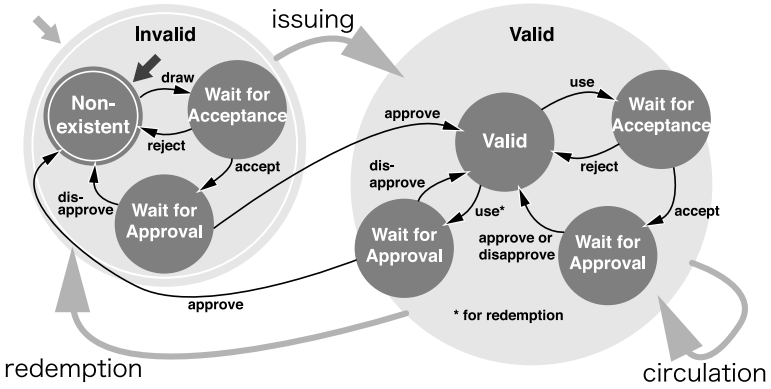
2.2 *i*-WAT: the Internet WAT System

Overview *i*-WAT is a translation of the WAT Core onto the Internet. We have made a reference implementation available, which has been used mainly by the members of the WAT System.

In *i*-WAT, messages signed in OpenPGP (*i-WAT messages*) are used to implement transfers of an electronically represented WAT ticket (*i-WAT ticket*).

An *i*-WAT ticket contains the identification number, amount of debt and public key user IDs of the drawer, users and recipients. Endorsements are realized by nesting PGP signatures. In our reference implementation, the chain of signatures is visualized as illustrated in Fig. 2, using the PGP photo IDs.

Table 1 shows the types of *i*-WAT messages. All *i*-WAT messages are signed by the senders, and are formatted in the canonical form[1] of XML[2] with nested signatures. The messages cause state transfers of a ticket as illustrated in Fig. 3.



* Gray arrows represent WAT state-transfer.
* Black arrows represent *i*-WAT state-transfer.

Fig. 3. State machine of a WAT/*i*-WAT ticket

Changes from the WAT System Upon translating the WAT Core onto the digital communication domain, we have made the following changes from the state machine of a WAT ticket:

1. Trades need to be asynchronously performed. Intermediate states, such as waiting for acceptance or approval, are introduced.
2. Double-spending needs to be prohibited. The drawer is made responsible for guaranteeing that the circulating ticket is not a fraud. This means that every trade has to be approved by the drawer of the involved ticket.

Protocol

Issuing – the birth of an i-WAT ticket

1. The drawer sends a <draw/> message which contains the public key user IDs of the drawer and lender, identification number and amount of debt. This message becomes the original *i*-WAT ticket after the protocol is completed.
2. The lender sends back the content of the message as an <accept/> message.
3. The drawer sends an <approve/> message to the lender.

Circulation – ordinary exchange

1. The user adds to the *i*-WAT ticket the public key user ID of the recipient, and sends it to the recipient as a <use/> message. This message becomes a valid *i*-WAT ticket after the protocol is completed.
2. The recipient forwards the content of the message to the drawer and user as an <accept/> message.

3. The drawer verifies the ticket, and sends an <approve/> message to the user and recipient.

Redemption – the return of the i-WAT ticket

1. The user sends a <use/> message to the recipient, who equals the drawer.
2. The drawer verifies the ticket, and invalidates it as the debt is now redeemed. The drawer sends an <approve/> message to the user.

Generalized Ticket Value We have recently made a generalization to the value of an *i*-WAT ticket such that it is expressed as a tetrad $\langle V_0, V_m, V_x, f \rangle$ presented by the drawer, where V_0 is the face value (initial value) of the ticket, V_m is the minimum value, V_x is the maximum value, and $f(t)$ is the differentiation (derivative) of a function of time $F(t)$. V_m/V_x are set to be $\perp/\top$ respectively if those values are not applicable.

The effective value V_t of a ticket at time t is given by the following equation:

$$V_t = \min(\max(\int_0^t f(t)dt + V_0, V_m), V_x)$$

This is a generalization to allow the value of a ticket to vary over time, limited by some minimum/maximum values. Typically, it holds that either $f(t) = 0$ for all t (*regular* ticket), $f(t) < 0$ for all t (*reduction* ticket) or $f(t) > 0$ for all t (*multiplication* ticket).

The incentive mechanism for reduction and multiplication tickets have been discussed in [14] and [13], respectively.

3 Trust Model

Fig. 4 shows the *trust model* of *i*-WAT, which is a definition of mutually validating relation $\overset{v}{\leftrightarrow}$, where A $\overset{v}{\leftrightarrow}$ B means that A and B validate the public keys of each other.

To implement the model by dynamically building an appropriate web of trust, [12] showed that it would suffice if the behaviors of participants satisfy the following three properties:

1. *mutual signing by knowing*, or any two mutual acquaintances sign the public keys of each other,
2. *mutual signing by participation*, or the drawer and a user of an *i*-WAT ticket sign the public keys of each other, and
3. *mutual full trust by participation*, or the drawer and a user of an *i*-WAT ticket fully trust each other, and a recipient fully trusts the corresponding user of a ticket, in the context of PGP public key signing.

Software features to help automating *mutual signing/full trust by participation* will be released in the near future.

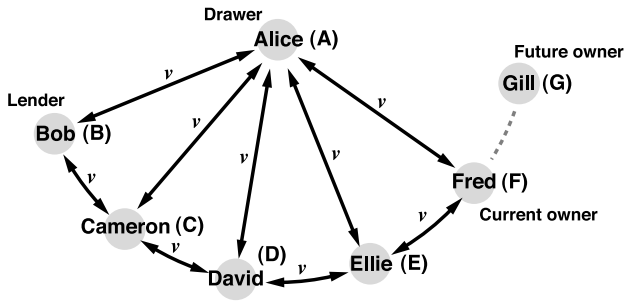


Fig. 4. *i*-WAT trust model

4 Incentive Model

We model a series of trades with an *i*-WAT ticket as a sequential game with incomplete information.

4.1 Notations and Preconditions

Participants Users are denoted as W (for WAT friends) indexed by the order of their appearance: drawer = W_0, lender = W_1, ..., current recipient = W_n. For the sake of argument, there assumed to be $n+1$ unique participants, and the webs of trust around them are built from scratch as transactions proceed.

Probability of Default Probability p_i devides W_i into two types: *successful* (appears by probability $1-p_i$) or *failing* (appears by probability p_i) to redeem the ticket in concern.

Timing of Usage The time at which W_i uses the ticket is regarded i to simplify reasoning. This means that the time is not evenly distributed in the model. Still, for any *reduction* tickets, it holds that $V_i < V_{i-1}$, and for any *multiplication* tickets, it holds that $V_i > V_{i-1}$, where $i > 0$.

Redemption takes place at time r.

Utility of Exchange There assumed to be some utility of having an exchange medium instead of having specific goods or unutilized services. This utility for W_i is denoted as UX_i.

UX_0 is a special case, where the value is divided into utility of spending UX_0^S and utility of earning (redeeming) UX_0^E, to reflect the fact that these events are not adjacent in the time line.

Cost of Trust Cost to rebuild trust relationships for W_i is CT_i. The cost includes that of *whitewashing*, or that one disappears and assumes a new identity. It is assumed that this cost does not vary in a large extent among participants, and is generally worth more than a value of a ticket. These assumptions should be justified by the fact that the *i*-WAT trust model requires construction of a *web of trust*[12], which requires that a new participant must know someone in person in the circle of friends around the *i*-WAT ticket.

Cost of lazy approval Cost of lazy approval by W_0 for a recipient W_i is denoted as CL_i. It is apparent that this cost exists for a *reduction* ticket, whose value is reduced over time. The cost exists for other types of tickets too, because it affects the usability of the ticket in concern; the ticket will not be usable by W_i until W_0 approves the transaction in which W_i received the ticket.

Laziness of W_0 is assumed to be observable from others. This assumption is justifiable by a software design; participants can observe how often W_0 becomes online in an *i*-WAT-enabled presence-sharing system.

Cost of premature redemption Cost of unexpectedly early redemption for W_0 is denoted as CP_0. Note that W_0 is incentivized to delay redemption even for *multiplication* tickets, which will often be used to control the timing of redemption by giving users incentives to wait.

Cost of communication Communication cost is negligible for *i*-WAT, which is the reason why the WAT System was electronized and made usable on the Internet.

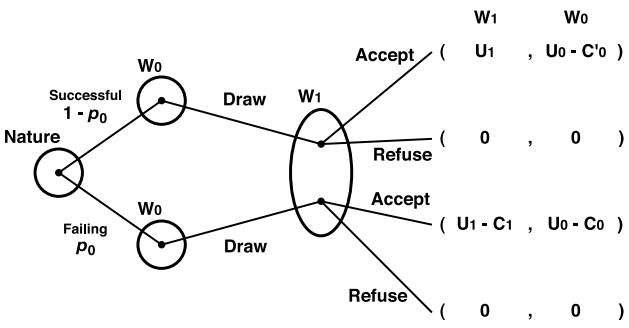
Accounting The sum of effective values of all tickets issued by W_0 in circulation is denoted as $\sum V$. This information is assumed to be made available to all prospective participants. Feasibility of this is discussed in section 6.

Since the cost of trust CT_0 is to be applied just once when W_0 whitewashes their identities, W_0 can minimize the effectiveness of the cost by issuing as many tickets as they can and then go on to default (see section 5.5). Therefore prospective lenders are interested in this information.

4.2 Game Trees

A *game tree* is a graph consisting of players' decision points as nodes, which are connected in the order of their occurrences. Each player has an *information set*, or a set of decision points from which they can choose an action. In the end of the graph, the gains of all players are drawn as leaves.

In the figures to follow, types of participants are not made explicit in the trees except for those of W_0, which are distinguished by probability p_0.



$$U_1 : UX_1 + V_1 - V_0 - CL_1, \quad C_1 : V_r(1 - p_1) + CT_1 p_1$$

$$U_0 : UX_0^S + V_0, \quad C_0 : \frac{V_r CT_0}{\sum V}, \quad C_0' : V_r + CP_0 - UX_0^E$$

Fig. 5. Game tree for issuing. $V_r = V_1$ and $p_1 = 0$ if W_1 is the last user.

Payoffs for issuing Fig. 5 shows a game tree for issuing an *i*-WAT ticket.

The first player is the nature who chooses between two types of W_0 as the drawer: *successful* or *failing* to redeem the ticket. These types appear by probabilities of $(1 - p_0)$ and p_0, respectively, for reasons either situational of strategic which are not distinguishable by other participants.

The lender W_1 has an information set in which the player is uncertain about W_0's type. Depending on the player's belief, W_1 chooses to either accept or refuse the ticket presented by W_0.

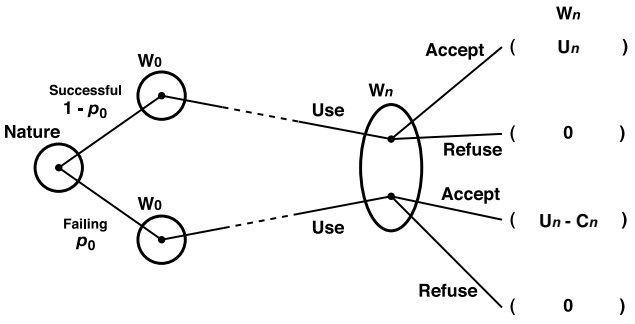
Inside parentheses are the gains of W_1 and W_0 in each combination of W_0's type and W_1's action.

1. If W_1 chooses to accept the ticket
   - W_1's expectation is $U_1 - C_1 p_0$
   - W_0's expectation is $U_0 - C_0'(1 - p_0) - C_0 p_0$
2. If W_1 chooses to refuse the ticket
   - Both W_0 and W_1 gain or lose nothing.

The utility UX_1 depends in large part on whether the ticket will be accepted by W_2 or not. It is also an important factor for minimizing $|V_1 - V_0|$ for a *reduction* ticket, in which case both W_0 and W_1 wish V_r to be zero. In case of a *multiplication* ticket, W_1 will typically wait until the effective value reaches V_x, and then use the ticket against W_0 for both maximizing their gain $V_1 - V_0$ (in case of successful W_0) and minimizing their loss to V_0 (in case of failing W_0).

In any case, p_0 is an important factor for W_1 to make a decision.

Payoffs for circulation Fig. 6 shows a game tree for circulating an *i*-WAT ticket. The tree is an extension to Fig. 5.



$$U_n : UX_n + V_n - V_{n-1} - CL_n, \quad C_n : (V_r(1-p_n) + CT_n p_n) \prod_{i=1}^{n-1} p_i$$

Fig. 6. Game tree for circulation. $V_r = V_n$ and $p_n = 0$ if W_n is the last user.

1. If W_n chooses to accept the ticket
   - W_n's expectation is $U_n - C_n p_0$
2. If W_n chooses to refuse the ticket
   - W_n gains or loses nothing.

If n is small, W_n is interested in the trustworthiness of all participants W_i where $0 \leq i < n$. Since $\prod_{i=1}^{n-1} p_i$ approaches zero as n increases, W_n will be indifferent of the type of W_0 if n is sufficiently large; they will tend to accept the ticket.

This may lead to a moral hazard, but still W_n will be interested in maintaining the trust model of *i*-WAT as described in the following section.

5 Protections against Moral Hazards

5.1 Overview

Table 2 shows the list of hazards in concern.

A case of someone receiving goods or service and escaping without providing a ticket is not discussed because it does not involve a successful *i*-WAT transaction, and there can be no proof of the incident within the context of the WAT Core (operational solutions need to be pursued).

Double-spending is also excluded from the list because its detection can be automated (it is in our reference implementation), and W_0 has no incentive to turn off such a software feature.

Table 2. Possible moral hazards and the imposed risks to the subjects

Name	Description	Risk to the Subject
Compromised secret	The subject's secret key is compromised or lost.	Cost of trust/Entrapment
Evidenceless signing	Signs public keys without checking their validity.	Impostors/Suspect for collusion
Evidenceless full trust	Gives full trust to someone without knowing them.	Impostors/Suspect for collusion
Excessive issuing	Issues an excessive amount of tickets.	Defaulting → cost of trust/ Premature redemptions
Lazy approval	Be late in approving transactions.	Premature redemptions
Defaulting	Defaults upon redemption.	Cost of trust
Empty promise	Receives the ticket and escapes without providing promised goods or service	Cost of trust

5.2 Sloppy Key Management

i-WAT uses public key cryptography as a protection against impostors. Failing to follow the good practice is considered a moral hazard. Keeping the good practice, on the other hand, maintains the trust model, and prevents offenders from getting away with unpaying the cost of trust.

This section describes how failing to follow the good practice in key management is against the subject's own interest. Discussions at later sections assume that the trust model is maintained.

Compromised Secret If a secret key is compromised or lost, the key needs to be declared invalid, and replaced with a new one. Since an *i*-WAT ticket records the public key user IDs[2] instead of the identifiers of the keys themselves, replacing the key does not affect the correctness of the data. However, this replacement costs equivalent to CT_i for W_i with the secret key in question because it involves reconstruction of the web of trust. Besides, the compromised key may be used for an entrapment (section 5.7).

Evidenceless Signing/Full Trust If participants sign public keys of others without personally validating them, or if they fully trust other participants without knowing their trustworthiness, there is a risk of allowing impostors of real or imaginary persons in the circle of friends around the *i*-WAT ticket.

Such impostors may perform misbehaviors like an empty promise, by which the signer/truster may be victimized. Or worse, they may be suspected as collaborators of such misbehaviors.

[2] A public key user ID is a character string. Under the current operation of PGP, it is typically an e-mail address.

5.3 Excessive Issuing

Excessive issuing can mean more debt than W_0 can handle, so that there is a risk of defaulting (increased p_0), which discourages both W_0 and W_1 to give birth to a ticket.

Furthermore, since excessive issuing is assumed to be observable from current ticket owners, they would want W_0 to redeem the tickets quickly, in order to avoid W_0's defaulting with the tickets they have. This should be especially true for those tickets whose chains of endorsements are still short. Which means that excessive and intensive issuing attracts premature redemptions.

5.4 Lazy Approval

There is a risk that circulation may be stalled by negligence of W_0 in their role of approving transactions.

Let us stand upon W_{n-1}'s view point. If W_0 is late to respond to the request for approval, the prospective transaction is delayed, costing CL_n to W_n which W_{n-1} knows that W_n can predict. Meanwhile, W_0 is not affected by their own laziness because acceptance and approval happen at the same time. When likelihood of acceptance is in question, W_{n-1}'s natural choise is to ask W_0 for redemption.

Therefore, being lazy is to risk premature redemptions, and W_0 is incentivized to respond quickly.

5.5 Defaulting

W_0 would want to minimize C_0 upon defaulting. If V_r can be reduced (as in the case of a *reduction* ticket), there may be no reason to default to begin with. Therefore, the only option for W_0 is to increase $\sum V$ to minimize the effect of CT_0. However, the value is monitored by all prospective lenders, so that W_0 cannot increase it over a reasonable amount.

5.6 Empty Promise

If there is a proof of an empty promise, W_0 can disapprove further transactions with the ticket. If the ticket has not been used further, W_{n-1} can safely become the valid owner of the ticket by a roll back.

The proof of the incident becomes a source of bad reputation for W_n, which can only be whitewashed by paying the cost of trust.

5.7 Collusions

There may be a colluded defaulting by every W_i where $0 \leq i < n$, so that W_n is victimized. However, the trust model implies that W_n must have needed to know someone in person in the chain of endorsement. At least that someone can be made to pay the cost of trust, which makes such collusion difficult.

There may be a colluded empty promise by W_0 and W_n so that W_{n-1} is victimized. This means that W_0 escapes too, in which case W_1 can take over the responsibility of the drawer. If it fails and the responsibility is forwarded upto W_{n-1}, it is indistinguishable from the state in which every W_i where $0 \leq i < n-1$ is colluding. The rest is the same as the case of a colluded defaulting.

Another form of colluding may be to entrap W_i so that it looks as if W_i committed a misbehavior such as an empty promise. This is only possible with a compromised secret key or a forged key pair, because there needs to be a verifiable signed message to prove that W_i did it. This requires a breach of the trust model.

6 Future Work

We have been implementing *i*-WAT as a plug-in for a messaging client called *wija*, which we are also developing. *wija* conforms to XMPP (Extensible Messaging and Presence Protocol)[4][5], and is available at the following URL:

– http://www.media-art-online.org/wija/

We intend to implement features to our software for monitoring excessive issuing: sharing information about tickets issued by others in circulation. We believe this can be done in a decentralized and trusted way. [11] briefly discusses a technique for doing this, which is an application of the protocol for *fair sharing* described in [9].

7 Related work

7.1 Geek Credit

Geek Credit[8] is an example of exchange medium usable on the Internet, which is close to *i*-WAT. It defines *Geek Credit policy*, which is similar to the *i*-WAT state machine, but the problem of double-spending is handled differently. Geek Credit detects double-spending at redemption, so that each trading does not need to be consulted with the drawer.

While this simplifies the protocol, the risk of attacks is higher for Geek Credit than for *i*-WAT. Recovery is also more difficult because the incident is only revealed at a later stage.

8 Conclusions

A medium of exchange which represents a guaranteed value should take an important role in the design of peer-to-peer systems, in which under-utilized resources are shared among selfish participants.

This paper showed that the design of i-WAT is incentive-compatible as to protection against moral hazards: taking advantage of the rules will result in the subject's confrontation to an uncontrollable risk, which is imposed by rational behaviors of other participants.

References

1. John Boyer. *Canonical XML Version 1.0*, March 2001. W3C Recommendation. Available electronically at http://www.w3.org/TR/xml-c14n.
2. Tim Bray, Jean Paoli, C.M.Sperberg-McQueen, and Eve Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*, October 2000. W3C Recommendation. Available electronically at http://www.w3.org/TR/REC-xml.
3. Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer. *OpenPGP Message Format*, November 1998. RFC 2440.
4. Peter Saint-Andre (Ed). *Extensible Messaging and Presence Protocol (XMPP): Core*, October 2004. RFC 3920.
5. Peter Saint-Andre (Ed). *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence*, November 2004. RFC 3921.
6. Joan Feigenbaum and Scott Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM '02)*, September 2002.
7. Paul Glover. Ithaca HOURs Online. Hypertext document. Available electronically at http://www.ithacahours.com/.
8. Alexander Komarov. Geek Credit homepage. Hypertext document. Available electronically at http://home.gna.org/geekcredit/.
9. T.-W. J. Ngan, D. S. Wallach, and P. Druschel. Enforcing fair sharing of peer-to-peer resources. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, California, February 2003.
10. Kenji Saito. Peer-to-peer money: Free currency over the Internet. In *Proceedings of the Second International Conference on Human.Society@Internet (HSI 2003), Lecture Notes in Computer Science 2713*. Springer-Verlag, June 2003.
11. Kenji Saito. Maintaining trust in peer-to-peer barter relationships. In *Proceedings of 2004 Symposium on Applications and the Internet (SAINT 2004 Workshops)*. IEEE Computer Society Press, January 2004.
12. Kenji Saito. WOT for WAT: Spinning the web of trust for peer-to-peer barter relationships. In *IEICE TRANSACTIONS on Communication*. The Institute of Electronics, Information and Communication Engineers, April 2005.
13. Kenji Saito, Eiichi Morino, and Jun Murai. Multiplication over time to facilitate peer-to-peer barter relationships. In *Proceedings of the 2nd International Workshop on P2P Data Management, Security and Trust (PDMST '05)*, August 2005 (to appear).
14. Kenji Saito, Eiichi Morino, and Jun Murai. Reduction over time: Easing the burden of peer-to-peer barter relationships to facilitate mutual help. In *Proceedings of the Second International Workshop on Computer Supported Activity Coordination (CSAC 2005)*, May 2005 (to appear).
15. Fritz Schwarz. Das experiment von Wörgl, 1951. Hypertext document. Available electronically at http://userpage.fu-berlin.de/∼roehrigw/woergl/, *(Shortened English translation by Hans Eisenkolb is available at http://www.sunshinecable.com/ ∼eisehan/woergl.htm)*.
16. Sidonie Seron. Local Exchange Trading Systems 1 - CREATION AND GROWTH OF LETS. Hypertext document. Available electronically at http://www.gmlets.u-net.com/resources/sidonie/ home.html.
17. watsystems.net. WATSystems home page. Hypertext document. Available electronically at http://www.watsystems.net/.