PAPER Special Section on Internet Technology V

WOT for WAT: Spinning the Web of Trust for Peer-to-Peer Barter Relationships

Kenji SAITO[†a)], ***Member***

SUMMARY Peer-to-peer complementary currencies can be powerful tools for promoting collaborations and building relationships on the Internet. *i*-WAT [1] is a proposed such currency based on WAT System [2], a polycentric complementary currency using *WAT tickets* as its medium 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 exchanges of messages signed in OpenPGP [3]. This paper clarifies the trust model of *i*-WAT, and investigates how it is related with that of PGP [4]. To implement the model by dynamically building an appropriate web of trust (WOT), we claim 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 trust the corresponding user of a ticket, in the context of PGP public key signing.

Likelihood of satisfaction of these properties is supported by the (dis)incentives imposed by the semantics of *i*-WAT. A reference implementation of *i*-WAT has been developed in the form of a Jabber [5] instant messaging client. We are beginning to put the currency system into practical use.

key words: *P2P, PGP, trust, currency, club formation*

1. Introduction

Distributed autonomous (or peer-to-peer) systems, such as an overlay network of people over the Internet, require coordination among participants to achieve their goals. Since each participant may behave selfishly to maximize their benefit, *incentive-compatibility* [6], roughly restated as the goal of the system being accomplished by collection of selfish behaviors, becomes important. Because relationships among participants in such a system necessitate fair exchanges of resources, the medium of exchange must take an important role.

Money is a well-known medium of exchange, but its scarcity has caused a lot of problems. *Complementary currencies*, or alternative forms of monetary medium, have been proposed and tested to achieve an autonomous, sustainable local economy even in short of money. There have been succeeding cases, such as experiments in Wörgl in 1932 (stamp money [7]), in Comox Valley in 1983 (Local Exchange Trading System [8]) and in Ithaca since 1991 (Ithaca HOURs [9]).

Many of the outcomes are short-lived, however, because most of the existing complementary currencies are dependent on the qualities of their administrations. It would thus benefit the autonomy and sustainability of economy if we could design an administration-free complementary currency; if we want to make a peer-to-peer world, money too needs to be peer-to-peer.

If such peer-to-peer complementary currencies are applied to the Internet, it would benefit many areas including multicast cost sharing, inter-domain routing, web caching, file sharing, distributed task allocation, and other application-layer overlay networks. Freedom to pursue these possibilities depends on whether we can have a free economic medium or not.

2. Background

2.1 Digital Signature

Digital signature is an essential technology for designing a dependable economic medium, which can provide a proof of debits or credits.

Throughout this paper, we use notations from [10] for formalization, with additional abstractions built upon them to fit our purposes.

Suppose Alice (A) is associated with a public/secret key pair denoted as $\langle K_A, K_A^{-1} \rangle$. To simplify the arguments to follow, we assume that each user has exactly one key pair associated with them.

A digital signature has two objectives:

1. To prove that Alice once admitted a message m.
2. To prove that m has not been altered since then.

These can be realized by encrypting m with Alice's secret key K_A^{-1}, obtaining $\{m\}_{K_A^{-1}}$ which is only decrypted with her public key K_A. Since K_A^{-1} is a secret known only to Alice, those who could decrypt $\{m\}_{K_A^{-1}}$ can infer that it must have been encrypted by Alice. They can also be certain that m has not been altered since Alice made $\{m\}_{K_A^{-1}}$ if the result of decryption equals m.

Usually, for efficiency reasons, instead of encrypting m itself, a digital signature is made by applying a secure hash function H to m, then encrypting the hash value with the secret key. H must be carefully chosen so that it is computationally infeasible to obtain m' where $m' \neq m$ such that $H(m) = H(m')$.

Manuscript received June 30, 2004.
Manuscript revised October 2, 2004.

[†]The author is with Graduate School of Media and Governance, Keio University, Fujisawa-shi, 252-8520 Japan.
a) E-mail: ks91@sfc.wide.ad.jp

DOI: 10.1093/ietcom/e88–b.4.1503

Definition 1 (digital signature): We write $A \overset{signs}{\rightarrow} m$ if and only if A presents both a plain-text message m and its encrypted form $\{H(m)\}_{K_A^{-1}}$. The latter is called a *signature* on the former.

The signature can be verified by Bob if he has a copy of Alice's public key K_A. To verify the signature, he calculates $H(m)$ from m, decrypts $\{H(m)\}_{K_A^{-1}}$ with K_A, and compares the two resulted values.

One question is how Bob can be sure that his copy of Alice's public key is genuine.

Definition 2 (validating relation): $x \overset{v}{\rightarrow} y$ if x possesses a copy of y's public key K_y, and infers that the copy is genuine.

We also write $x \overset{v}{\leftrightarrow} y$ iff $x \overset{v}{\rightarrow} y \wedge y \overset{v}{\rightarrow} x$ (*mutually validating relation*).

A trust model around validity of public keys is a specific definition of *validating relation* $\overset{v}{\rightarrow}$ in the system in concern. Typically, validity of a public key is supported by a *certificate*, or a signature on the key. For example, if Bob (B) sees Cameron (C) such that $B \overset{v}{\rightarrow} C \wedge C \overset{signs}{\rightarrow} K_A$, then $B \overset{v}{\rightarrow} A$ assuming that C's certificate is trustworthy. This relation is recursive, so that someone needs to self-certify at some point.

A public key infrastructure uses a tree of *certificate authorities*, or issuers of certificates, whose public keys are validated by the parent nodes, rooted by a self-certifying authority.

2.2 Web of Trust

In a *web of trust*, however, responsibility for validating public keys is delegated to people one trusts, without necessitating certificate authorities. It is a network of people signing the public keys of others.

Signing relation $\overset{s}{\rightarrow}$ states that one certifies that its copy of someone's public key is genuine.

Definition 3 (signing relation): $\overset{s}{\rightarrow}$ is defined as follows:

1. $x \overset{s}{\rightarrow} x$
2. $x \overset{s}{\rightarrow} y$ if $x \overset{signs}{\rightarrow} K_y$

We also write $x \overset{s}{\leftrightarrow} y$ iff $x \overset{s}{\rightarrow} y \wedge y \overset{s}{\rightarrow} x$ (*mutually signing relation*).

Definition 4 (signing-apart relation): $\overset{s[n]}{\rightarrow}$ is defined as follows:

1. $x \overset{s[0]}{\rightarrow} x$
2. $x \overset{s[1]}{\rightarrow} y$ if $x \overset{s}{\rightarrow} y \wedge x \neq y$.
3. $x \overset{s[a+b]}{\rightarrow} z$ if $x \overset{s[a]}{\rightarrow} y \wedge y \overset{s[b]}{\rightarrow} z$.

We also write $A \overset{s}{\rightarrow} B \overset{s[n]}{\rightarrow} C$ in place of $A \overset{s[n+1]}{\rightarrow} C$ if $A \overset{s}{\rightarrow} B \wedge B \overset{s[n]}{\rightarrow} C$ (expansion of signing-apart relation) in order to clarify who stands in between the chain of signing relations.

Definition 5 (web of trust): A web of trust for x is a set of all y such that $x \overset{s[n]}{\rightarrow} y$ where $n \geq 0$.

A specific validation relation needs to be defined over a web of trust. PGP (Pretty Good Privacy) is an example of a cryptographic technology which defines such a relation. We use GnuPG [11] as our choice of implementation of OpenPGP [3] standard.

2.3 PGP Trust Model

Let us further define that $\mathcal{T}_x$ is the set of users x considers fully trustable, and $\mathcal{T}'_x$ is the set of users x considers marginally trustable.

In the context of PGP public key signing, *fully trustable* means that one considers that the owner of a public key has an excellent understanding of key signing, and his or her signature on a key would be as good as their own, and *marginally trustable* means that one considers that the owner of a public key understands the implications of key signing and properly validates keys before signing them [4].

The PGP trust model is a definition of *validating relation* $\overset{v}{\rightarrow}$ over a web of trust[†].

Definition 6 (PGP trust model): $x \overset{v}{\rightarrow} y$ if

1. sufficient number of valid key owners sign y's public key, i.e.
   a. $x \overset{s}{\rightarrow} y$, or
   b. there exist at least f instances of z such that $z \in \mathcal{T}_x$, $x \overset{v}{\rightarrow} z \wedge z \overset{s}{\rightarrow} y$, or
   c. there exist at least m instances of z such that $z \in \mathcal{T}'_x$, $x \overset{v}{\rightarrow} z \wedge z \overset{s}{\rightarrow} y$; and
2. $x \overset{s[n]}{\rightarrow} y$ where $n \leq h$,

where f, m and h are the required number of fully trusted key owners, required number of marginally trusted key owners, and number of maximum steps in the path in the web of trust tracing x back from y, respectively.

By default, GnuPG defines $f = 1$, $m = 3$ and $h = 5$.

2.4 WAT System

2.4.1 Overview

WAT System [2] is a complementary currency designed by Mr. Eiichi Morino, the founder of Gesell Research Society Japan [12]. 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

[†]PGP also allows marginal validation of public keys, which is not used in the design of our currency.

trade (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[†], the present date, and the drawer's signature. The drawer gives the ticket to the lender, and in return obtains the goods or service.
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.

Figire 2 shows the state machine of a WAT ticket.

2.4.2 Distinctive Features

(1) Autonomy

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

(2) 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.

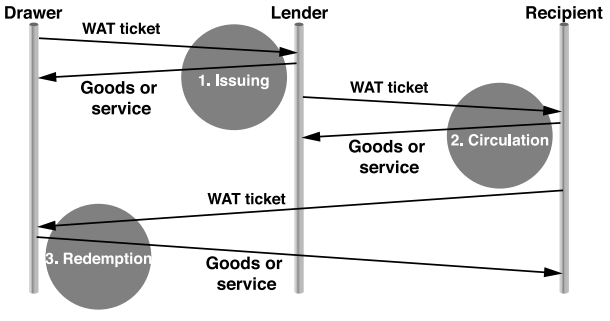

Fig. 1 Three stages of trading with a WAT ticket.

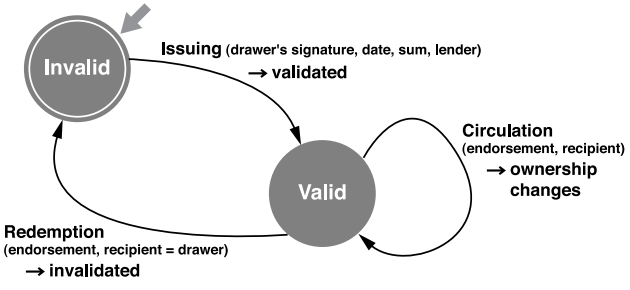

Fig. 2 State machine of a WAT ticket.

(3) Extensibility

The protocol illustrated in Fig. 1, 2 defines *WAT Core*, the essence of WAT System. An *extended part* can be defined for a new currency based on WAT System, stating, for example, the region, group and duration in which the tickets are usable, as well as the unit in which the debit is quantified.

(4) Security

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

3. *i*-WAT: The Internet WAT System

3.1 Overview

i-WAT is a translation of WAT Core onto the Internet. In *i*-WAT, messages signed in OpenPGP are used to implement transfers of an electronically represented WAT ticket. The exchanged messages are called *i-WAT messages*, and the ticket represented by the messages is called an *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 as illustrated in Fig. 3.

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 [13] of XML [14] with nested signatures. The messages cause state transfers of an *i*-WAT ticket as illustrated in Fig. 4.

3.2 Changes from WAT System

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

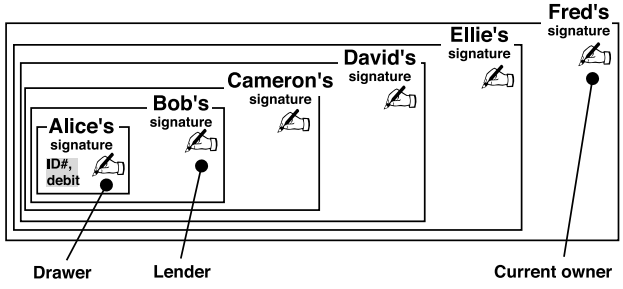

Fig. 3 Signature chain in an *i*-WAT ticket.

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

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.

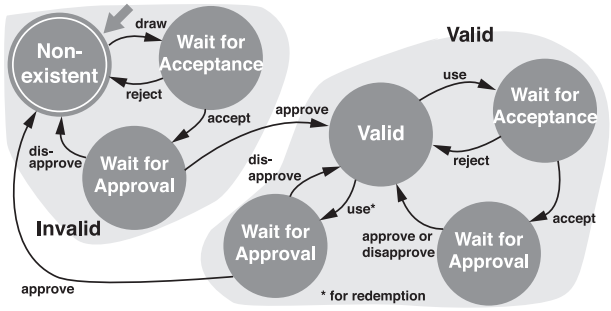

Fig. 4 State machine of an *i*-WAT ticket.

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.

3.3 Protocol

3.3.1 Issuing—The Born 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.

3.3.2 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.

3.3.3 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 debit is now redeemed. The drawer sends an <approve> message to the user.

4. *i*-WAT and the PGP Trust Model

4.1 *i*-WAT Trust Model

Let us define that $t(x)$ is an *i*-WAT ticket t drawn by x, $\mathcal{U}_{t(x)}$ is the set of users throughout the lifecycle (up to redemption) of $t(x)$, and $y \overset{t(x)}{\rightarrow} z$ denotes that y gives $t(x)$ to z as a result or promise of a trade.

The *i*-WAT trust model is a definition of *mutually validating relation* $\overset{v}{\leftrightarrow}$ over a network of participants.

Definition 7 (*i*-WAT trust model): for every $t(x)$,

1. for all y such that $y \in \mathcal{U}_{t(x)}$, $x \overset{v}{\leftrightarrow} y$
2. for all y, z such that $\{y, z\} \subseteq \mathcal{U}_{t(x)}$, $y \overset{v}{\leftrightarrow} z$ if $y \overset{t(x)}{\rightarrow} z$

Fig. 5 illustrates the model by an example. This model is naturally induced from the necessity for the participants to validate *i*-WAT messages.

4.2 Spinning the Web of Trust—Preconditions

If the PGP trust model over the network of participants does not readily support the above model, the model needs to be implemented by dynamically building an appropriate web of trust. In order to do so, we claim that it suffices (but not necessitates) if the behaviors of the participants satisfy the following properties.

Property 1 (mutual signing by knowing):
for every x and y,

- $x \overset{s}{\leftrightarrow} y$ if x knows[†] y

Intuitively, this states that any two mutual acquaintances sign the public keys of each other.

Property 2 (mutual signing by participation):

[†]In the context of this paper, *knows* relation is defined to be symmetrical, i.e., y knows x if x knows y.

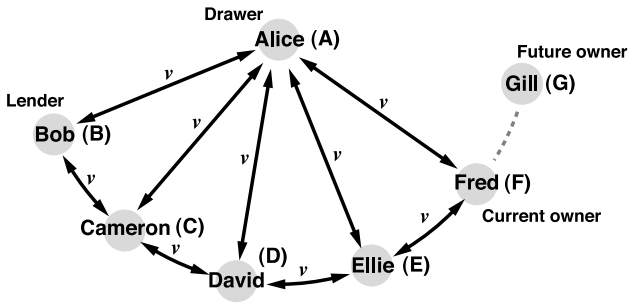

Fig. 5 *i*-WAT trust model.

for every $t(x)$,

- for all y such that $y \in \mathcal{U}_{t(x)}$, $x \overset{s}{\leftrightarrow} y$

Intuitively, this states that the drawer and a user sign the public keys of each other.

Property 3 (mutual full trust by participation):
for every $t(x)$,

1. for all y such that $y \in \mathcal{U}_{t(x)}$, $x \in \mathcal{T}_y \wedge y \in \mathcal{T}_x$
2. for all y, z such that $\{y, z\} \subseteq \mathcal{U}_{t(x)}$, $y \in \mathcal{T}_z$ if $y \overset{t(x)}{\rightarrow} z$

Intuitively, this states that the drawer and a user are confident about each other, and a recipient is confident about the corresponding user, that their correspondents have an excellent understanding of key signing. They need to reflect such views in their PGP trust databases.

Also we assume that GnuPG's default values are used for variables f, m and h.

4.3 Spinning the Web of Trust—Case Studies

We justify the above claim by case studies. Throughout the studies, the network of participants in Fig. 5 is used as an example. It is assumed that no external source of information is available.

Our goal is to show that the *i*-WAT trust model is satisfied in every stage of trades starting from likely initial states, i.e., a joining party knows someone in the network of participants, if the properties explained in Sect. 4.2 are satisfied by the participants.

The statement that follows each claim is both a casual proof and a procedure to achieve the goal.

4.3.1 Issuing

The goal is to form the initial network of participants between Alice and Bob.

Claim 1: $A \overset{v}{\leftrightarrow} B$ results if and only if Alice knows Bob.

(1) In case Alice knows Bob

1. By *mutual signing by knowing*,
$$A \overset{s}{\leftrightarrow} B$$
2. By the definition 1a of *PGP trust model*,
$$A \overset{v}{\leftrightarrow} B$$

(2) In case Alice does not know Bob

There is no definition or property available to deduce $A \overset{v}{\leftrightarrow} B$.

4.3.2 Circulation

The goal is to let Gill join the existing network of participants.

Claim 2: $G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$ results if Gill knows either Fred or Alice, or someone (in $\mathcal{T}_G$) or some people (in $\mathcal{T}'_G$) in the network of participants.

(1) In case Gill knows both Fred and Alice

1. By *mutual signing by knowing*,
$$G \overset{s}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} A$$
2. By the definition 1a of *PGP trust model*,
$$G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$$

(2) In case Gill knows Fred, but not Alice

1. By *mutual signing by knowing* and *mutual signing by participation*,
$$G \overset{s}{\leftrightarrow} F \wedge F \overset{s}{\leftrightarrow} A$$
2. By *expansion of signing-apart relation*,
$$G \overset{s}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} F \overset{s}{\leftrightarrow} A$$
3. By the definition 1a of *PGP trust model*,
$$G \overset{v}{\leftrightarrow} F \wedge G \overset{s}{\leftrightarrow} F \overset{s}{\leftrightarrow} A$$
4. $F \in \mathcal{T}_A$ and $F \in \mathcal{T}_G$ by the properties 1 and 2 of *mutual full trust by participation*, respectively. Also, the path length between Gill and Alice is shorter than h. Therefore, by the definition 1b of *PGP trust model*,
$$G \overset{v}{\leftrightarrow} F \wedge G \overset{v}{\leftrightarrow} A$$

(3) In case Gill knows Alice, but not Fred

1. By *mutual signing by knowing* and *mutual signing by participation*,
$$G \overset{s}{\leftrightarrow} A \wedge A \overset{s}{\leftrightarrow} F$$
2. By *expansion of signing-apart relation*,
$$G \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$
3. By the definition 1a of *PGP trust model*,
$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

4. $A \in \mathcal{T}_G$ and $A \in \mathcal{T}_F$ by the property 1 of *mutual full trust by participation*. Also, the path length between Gill and Fred is shorter than h. Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{v}{\leftrightarrow} F$$

(4) In case Gill knows neither Alice nor Fred

The goal can still be met if

1. there is one user x such that $x \in \mathcal{U}_{t(A)}$ who knows Gill and appeared earlier than Fred, and $x \in \mathcal{T}_G$, or
2. there are three users x, y, z such that $\{x, y, z\} \subset \mathcal{U}_{t(A)}$ who all know Gill and appeared earlier than Fred, and $\{x, y, z\} \subseteq \mathcal{T'}_G$.

The proofs for the above two cases are similar; they both involve first establishing $G \overset{v}{\leftrightarrow} A$ by way of someone or some people in the middle, only that the latter is more complex.

Suppose Gill knows Cameron, David, Ellie, and marginally trust them.

1. By *mutual signing by knowing* and *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} C \wedge C \overset{s}{\leftrightarrow} A \wedge A \overset{s}{\leftrightarrow} F$$

2. By *expansion of signing-apart relation*, and by the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} C \wedge C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

3. The above also holds if we replace C with D or E. It is given that $\{C, D, E\} \subseteq \mathcal{T'}_G$. Also, the path length between Gill and Alice is shorter than h. Therefore, by the definition 1c of *PGP trust model*,

$$G \overset{v}{\rightarrow} A \wedge C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

4. $C \overset{v}{\leftrightarrow} A$ by the definition 1a of *PGP trust model*. $C \in \mathcal{T}_A$ by the property 1 of *mutual full trust by participation*. Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} C \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

5. Now that Gill and Alice mutually validates their public keys, they can establish $G \overset{s}{\leftrightarrow} A$ by *mutual signing by participation*.

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{s}{\leftrightarrow} A \overset{s}{\leftrightarrow} F$$

6. $A \in \mathcal{T}_G$ and $A \in \mathcal{T}_F$ by the property 1 of *mutual full trust by participation*. Also, the path length between Gill and Fred is shorter than h. Therefore, by the definition 1b of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A \wedge G \overset{v}{\leftrightarrow} F$$

4.3.3 Redemption

The goal is to complete the lifecycle of the ticket in concern without further expanding the existing network of participants.

Claim 3: $G \overset{v}{\leftrightarrow} A$ results without building the web of trust any further.

1. By *mutual signing by participation*,

$$G \overset{s}{\leftrightarrow} A$$

2. By the definition 1a of *PGP trust model*,

$$G \overset{v}{\leftrightarrow} A$$

4.4 Justification of the Preconditions

We casually explain how the preconditional properties are supported by the natural behaviors of people. The formal proof that the design of *i*-WAT is incentive-compatible is left out for a future work.

4.4.1 Mutual Signing by Knowing

If two parties know each other (well enough), it should be possible to safely exchange the fingerprints[†] of their public keys. Therefore this is only a question of the communication cost.

4.4.2 Mutual Signing by Participation

Because it becomes easier for other people to join the circle of friends around an *i*-WAT ticket if this property is met, both the drawer and user have incentives to sign each other's public keys after properly validating them.

4.4.3 Mutual Full Trust by Participation

The participants are motivated to fully trust their correspondents in the context of public key signing by the same incentives as the above. Also, they are disincentivized to be negligent of the precautions for signing public keys, in order to protect themselves from possible attacks by impostors.

5. Deployment

5.1 Overview

i-WAT allows the underlying carrier of messages to be existing e-mail or instant messaging systems. As a reference implementation, we have developed an *i*-WAT plug-in and the hosting Jabber [5] client called *wija*. The software is available from http://www.media-art-online.org/wija/ (the *i*-WAT

[†]A fingerprint is a hash value of a key so that the key's identity can be checked with small cost.

plug-in is bundled with all platform-specific packages).

i-WAT, as well as public key exchange to support the system, have been implemented as extensions to Jabber instant messaging protocol.

The reference implementation has already been in use by the WAT System communities. It has been used, for example, to exchange goods, such as books, with services, such as working hours for developing an open source software, namely *wija* itself.

5.2 Support for the Preconditional Properties

Our software lets users exchange their public keys directly (by way of Jabber servers) without consulting a public key server. From a user's point of view, this is performed by choosing a correspondent from a buddy list, and selecting either importing or exporting their keys. When imported, a window pops up with the fingerprint of the public key, asking the user whether to sign the key or not. This is expected to enhance the ease for *mutual signing by knowing*.

Our software currently does not directly support *mutual signing by participation*. However, the current design of *wija* and its *i*-WAT plug-in uses the buddy list to locate the owner of a public key, so that new participants will be required to add the drawer in their buddy list if they have not already, to which the drawer would respond by adding them back. Then the above mechanism for key exchange can be used.

Our software currently does not have a support for *mutual full trust by participation*.

6. Future Work

We will add a user interface to *wija* to support *mutual signing* and *mutual full trust by participation*. We will experiment further how we can reduce the communication cost so that people can easily satisfy the three preconditional properties. At the same time, as we put *i*-WAT into practical use, we see if these properties are actually useful for building up the circle of friends around an *i*-WAT ticket.

7. Related Work

7.1 Magic Money

Magic Money [15] is an example of message-based currencies on the Internet based on PGP signatures. It was designed and implemented by an anonymous programmer known as Pr0duct Cypher in the early 1990s. Although there were a few enthusiasts, the use of Magic Money did not spread widely for several reasons:

1. It utilized Chaum's blind signature protocol [16] which was patented at the time. Since Magic Money was distributed as a free, open source software, its existence itself was unlawful.
2. It required presence of a server, which had to be maintained by someone.
3. It pursued untraceability while there was nothing to back up the values of the digital coins. The system was regarded as untrustworthy.

We regard Magic Money as an important experience of deploying a complementary currency on the Internet, and have tried to do the opposites: 1) we chose GnuPG as the implementation of OpenPGP which does not use patented technologies, 2) we chose not to rely on servers (we use Jabber servers as routers), and 3) we chose to give up anonymity to some extent (public key IDs and signing relations are made known) to build up trust instead.

7.2 Geek Credit

Geek Credit [17] is an example closer 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. Having not to consult the drawer also makes the trust model of Geek Credit simpler, but it means that there is no implicit support for building the web of trust dynamically other than joining the circle of friends by knowing the current owner of the ticket. Since the drawer does not have a way to check the usage of their tickets, there is no way to enforce the imposed restrictions by an extended part if there is one.

8. Conclusions

Peer-to-peer complementary currencies can be powerful tools for promoting collaborations and building relationships on the Internet. *i*-WAT is a proposed such currency based on WAT System, a polycentric complementary currency using WAT tickets whose values are supported by chains of trust.

In this paper, we clarified the *i*-WAT trust model. To implement the model by dynamically building an appropriate web of trust, we showed that it would suffice if the behaviors of participants satisfy the following three properties:

1. *mutual signing by knowing*
2. *mutual signing by participation*
3. *mutual full trust by participation*

Likelihood of satisfaction of these properties is supported by the (dis)incentives imposed by the semantics of *i*-WAT.

We have developed a Jabber client called *wija* in order to put *i*-WAT into practical use. We have been experimenting on user interfaces for exchanging publice keys, so that participants of *i*-WAT can satisfy the above properties with little or no subjective communication cost.

Acknowledgement

The author would like to thank Mr. Eiichi Morino and other

members of Gesell Research Society Japan for valuable advices and discussions on the content of this paper, as well as feedback on *wija* software.

References

[1] K. Saito, "Peer-to-peer money: Free currency over the Internet," Proc. Second International Conference on Human.Society@Internet (HSI 2003), Lecture Notes in Computer Science 2713, Springer-Verlag, June 2003.
[2] watsystems.net, "WATSystems home page," Hypertext document. Available electronically at http://www.watsystems.net/
[3] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, "OpenPGP message format," RFC 2440, Nov. 1998.
[4] The Free Software Foundation, "The GNU privacy handbook," Available electronically at http://www.gnupg.org/
[5] J. Miller, "XMPP instant messaging," Internet-Draft, Jan. 2003.
[6] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," Proc. 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM'02), Sept. 2002.
[7] F. Schwarz, "Das experiment von Wörgl," Hypertext document, 1951. 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).
[8] S. 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
[9] P. Glover, "Ithaca HOURs online," Hypertext document, Available electronically at http://www.ithacahours.com/
[10] M. Burrows, M. Abadi, and R.M. Needham, "A logic of authentication," Proc. Royal Society, vol.426, no.1871, 1989.
[11] The Free Software Foundation, "The GNU privacy guard," Hypertext document. Available electronically at http://www.gnupg.org/
[12] Gesell Research Society Japan, Hypertext document, Available electronically at http://www.grsj.org/
[13] J. Boyer, Canonical XML Version 1.0, March 2001. W3C Recommendation, Available electronically at http://www.w3.org/TR/xml-c14n
[14] T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler, Extensible Markup Language (XML) 1.0 (Second ed.), Oct. 2000. W3C Recommendation. Available electronically at http://www.w3.org/TR/REC-xml
[15] M. Woodcock, "Magic Money," Hypertext document, Available electronically at http://www.csee.umbc.edu/~woodcock/cmsc482/proj1/magmoney.html
[16] D. Chaum, "Blind signatures for untraceable payments," Advances in Cryptology—Crypto'82, pp.199–203, Springer-Verlag, 1983.
[17] A. Komarov, "Geek Credit homepage," Hypertext document. Available electronically at http://home.gna.org/geekcredit/

Kenji Saito was born in 1964. He received M.Eng. in Computer Science from Cornell University in 1993. Areas of his interest include real-time systems and distributed autonomous systems.