Incentive Mechanisms in P2P Networks: Micropayment Systems

Sánchez F. * **Cela A.** **

* *Universidad Carlos III de Madrid, Facultad de Ingeniería Telemática*
Madrid, España (e-mail: franklin.sánchez@alumnos.uc3m.es)
** *Escuela Politécnica Nacional, Facultad de Automatización y Control*
Quito, Ecuador (e-mail: andres_cela@ieee.org)

Resumen: *En la actualidad se ha extendido ampliamente el uso de redes P2P, particularmente en soluciones para compartir contenidos; sin embargo, sus ventajas de rendimiento y escalabilidad se han visto comprometidas debido al comportamiento egoísta de sus miembros. Por esta razón, varios autores han propuesto mecanismos de incentivo económico para recompensar a aquellos miembros que comparten recursos o contenido a la red. Varios sistemas de micropago han sido propuestos con esta finalidad, y en general buscan tomar ventaja de las características de las redes P2P para maximizar su eficiencia.*
En este artículo, se presenta una revisión de varios de los sistemas de micropago propuestos hasta la actualidad y se los analiza en base a cuatro criterios clave: escalabilidad, transferibilidad, seguridad y anonimato. Esto, como parte inicial de un estudio del impacto de estos mecanismos en este tipo de redes.

Palabras clave: *Sistemas de micropago, Redes P2P, moneda transferible, seguridad, anonimato.*

Abstract: *Today the use of peer-to-peer networks has been widespread; particularly in content sharing solutions; however its performance and scalability advantages are compromised due to the selfish behavior of its members. This is why authors have proposed financial incentive mechanisms to reward peers who share resources or content to the network. Several micropayment systems have been proposed for this purpose, and in general they look to take advantage of the P2P networks characteristics to maximize their efficiency.*
In this paper, we present a review of several micropayment systems proposed to date and we analyze them based on four key criteria: scalability, transferability, security and anonymity. This, as initial part of a study of the impact of these mechanisms in this kind of networks .

Keywords:*Micropayment systems, P2P Networks, transferable coin, security, anonymity.*

1. INTRODUCTION

All of us know the popularity and importance that the P2P networks have acquired in recent years as one of the best solutions for collective sharing of content, in such networks, each of its members can freely exchange resources and/or content without the intervention of a central server, making P2P networks highly scalable and with high performance since each network member bring their computing resources to it.

The performance of these networks is based on the voluntary contribution of resources and/or content by each individual participant, and since, no participant receives some additional benefit for contribute to the network, these may have a "free-riding behavior" where certain peers benefit from network resources without contributing anything to it. This behavior ultimately creates vulnerabilities in the network and is detrimental to the overall performance of it.

In 2000, studies were conducted on "free-riding behavior" over the *Gnutella* network in which was found that over 70% of peers take advantage of the benefits of the network without contributing to it and 90% of them did not respond to the queries of the peers [1].

Later, Golle et al. [7] proposed incentives for sharing, introducing the concept of micropayments for peer-to-peer networks, which is nothing other than the payment of small amounts of money for each service that a peer offers to the network.

Many micropayment schemes as Millicent [6] and MicroiKP [8] or PayWord [13] require a broker or central server that is responsible for checking all transactions giving to an "on-line" service, and transactions can only be done between a specific client and server relationship, which make this type of solutions not scalable and therefore not suitable for use in P2P networks, since none of them take advantage of the main features of the peer-to-peer networks such as [15]:

- Peers can play the role of customers (buyers) or servers (sellers), so it would be appropriate to use a currency that can be transferred between them without the intervention of a central broker.
- Each peer has a large amount of resources that can be shared on the network.

Yang et al. proposed in 2003 PPay [15], which would be one of the first micropayment schemes designed to exploit the potential of the peer-to-peer networks and beyond many other schemes have been proposed, and it is our goal to make a review of the main protocols proposed as incentive mechanisms for sharing in P2P networks as initial part of a study of the impact of these mechanisms in this kind of networks.

2. MICROPAYMENT SYSTEMS ALGORITHMS

2.1 CPay: A new micropayment protocol based on P2P Networks

CPay [10] like previous works as PPay [15] uses transferrable coins, but it differs because it exploits the heterogeneity of peers. In this scheme, for each transaction a customer peer can compute a *consistent hashing function* in order to find an *eligible peer* called *Broker Assistant (BA)* to check the coin and decide whether to authorize the transaction or not. In this manner the broker is only responsible for selling coins, paying peers and managing these eligible peers.

The dynamic *consistent hashing* used in CPay, maps a peer from the set of all peers to a *Broker Assistant (BA)*, which is part of the subset of high performance peers. CPay defines dynamic consistent hashing as a family of hash functions $f_y : X \to Y$, where the set of Y can change dynamically, and satisfies the following two conditions:

1. There exists a very small value $\forall\ y \in Y, \left|\{X|f_y(X) = y\}\right| \leq \varepsilon \times |X|/|Y|$; meaning that the elements in X are almost evenly mapped to the elements in Y.
2. By adding an element to or deleting an element from the set of Y, we can get a new set of Y', and for any Y and Y', $|\{x|f_Y(X) \neq f_{Y'}(X)\}| < O(|X|/|Y|)$; meaning that when adding an element to or deleting an element from Y, only a small number of elements in X need to change their mappings.

A. Participating Parties

In CPay the broker is only responsible for coins distribution and redemption, and the management of eligible peers, so it does not need to participate in the transaction; therefore, there are only three parties involved in a transaction:

1. *Payer*, the peer that wants to pay in order to get a service.
2. *Payee*, the service provider that wants money in return for your goods.
3. *Broker Assistant BA*, the eligible peer which the payer is mapped to and is responsible for checking the coin and the authorization of the transaction. BA can also be a payer or a payee; in which case, $BA's$ transactions should be handled by the $BA's$ mapping BA. The consistent hashing function must guarantee that no one will be mapped to itself.

B. Transaction procedures

1. *Coin Purchase*: A peer can get coins in two ways: buying coins to the broker or through another peer by selling services, so that money can be transferred in two ways:
   a. As *Coin* $C = (BA'_U, U, SNO)_{SBroker}$, where SNO is the unique identifier of the coin and BA'_U is the mapping BA of U when the broker generates the coin, and the broker's signature is present (when a peer bought a coin from the broker).
   b. As an *Authorization Message*.
2. *Request:* It represents a requesting message that has the form $Re = (X, U, Au)_{SX}$, and it is sent by X to payer $X's$ mapping BA, which is indicated in Au. This message indicates that payer X requests an authorization to pay the coin C indicated in Au to payee U.
3. *Authorization:* This message is sent to the payee U by the payer's mapping $BA(BA_X)$ and indicates that BA_X authorizes payer X to pay the coin C to the payee U. Authorization has the form $Au = (BA'_U, U, TS, C, BA_X)_{s_B A_X}$. This message also consist of time stamp TS which indicates the time when this authorization has happened.
4. *Request:* When peer U wants to spend the same coin issued by peer X to peer V. It sends a request message to BA_U asking for authorization.
5. *Authorization:* This message is sent to payee V by BA_U and indicates that BAu authorizes payer U to pay the coin C to the payee V.

2.2 P2P NetPay

P2P NetPay [2] is an off-line debit-based scheme based on the client-side e-wallet NetPay protocol [4], which in turn is based on PayWord protocol [13].

Three main elements are considered in this scheme: The peer consumer/user C, the vendor V, and the broker B. The system also assumes that broker is honest and trusted by both the customers and vendors, while the customer and vendor may be or not honest.

Brokers, users, and vendors use public/secret key pair to sign their messages; thus, a message M signed by a secret key SK is denoted $\{M\}_{SK}$ and can be verified using the corresponding public key PK.

As it is discussed below, this scheme uses a *Touchstone* signed by the *broker* to ensure the integrity and security of *e-coins* and an *Index* to prevent double spending.

A. Overview

In the P2P-NetPay architecture, initially a customer accesses the broker's web site to create an account; in this registration process each peer receives a pseudonym (ID_C), which will be used as an identity during transactions; thereby ensuring that only the secure broker can identify the participants in a particular transaction. Then the customer needs to buy a number of e-coins from the broker. This process occurs in a called *Customer-Broker Transaction* [5].

When the customer ask for e-coins, the broker does two actions:

1. Debit money from the customer account and creates a payword chain (which represent a set of e-coins) $w_0, w_1, w_2, \ldots, w_n, w_{n+1}$ by computing
$$w_i = h(w_{i+1})$$
where $i = n, n-1, n-2, \ldots, 0$ and h is the MD5 hash function [12.1]. Then the broker sends to the customer a message with the payword chain encrypted with the customer's public key.
$$M_2 = \{w_1, w_2, \ldots, w_n\}_{PK_C}$$
To prevent that the customer overspends and forges paywords, the broker keeps the seed $w_n + 1$.
2. Computes the *touchstone* (the root element w_0 of a payword chain) for that chain
$$T = \{ID_C, w_0\}_{SK_B}$$

When a customer makes a purchase from a vendor V_1, he sends a message M_3 to the vendor containing the service required, the payment P, and the IP of the Broker. The price for the service should be agrees between the parts.

$$M_3 = \{Service, P, IP_B\}$$

where the payment P are the e-coins. Then the vendor requests by a message the *touchstone* and *index* from the broker, and with this information the payment P is verified by the vendor (coin is verified and sufficient credit remains). If the payment is valid V_1 provides the services to C.

If C wants to purchase services from another vendor V_2, once the service request has been made, V_2 requests the current touchstone and index information from V_1; V_1 signs the index and sends the information to V_2

$$M_4 = \{T, index\}$$

Then, V_2 verifies e-coins and debit coins (increase index) to further provide the service. If V_1 is offline when V_2 request the touchstone, V_2 can contact the broker in order to get it, this is possible because V_1 must transfer touchstone and index to the broker before he goes down.

At the end of the day, each vendor sends the e-coins to the broker in order to redeem them with real money. The broker verifies each payword by performing hashes on it and counting the amount of paywords. If the paywords are correct, the broker deposits the amount on the vendor's account, and an acknowledgment is sent to the vendor.

2.3 A Secure and Lightweight Micropayment Scheme in P2P Networks

The scheme proposed in [9] has a registry server, which is responsible for manage the admission of nodes to enter into the system. To join the system, each peer should register, and then gets a signed account and the public system's key. The account contains an initial amount of tokens and is signed by a quorum of *Trusted Nodes (TN)*, which poses one part of the system's private key, the account information can be kept by the node himself. During each transaction, TN will update both, the payer and payee's account information.

In order to avoid the use of expired accounts the system uses *Account Holders AH* to maintain up to date the account information. AH does not keep detailed information about accounts, AH only keep ID, time and token quantity of the account. Then there are five parties involved in the system: *Registry Server, Consumer, Provider, Trusted Nodes*, and *Account Holders*.

Trusted Nodes are a quorum of nodes which posses one part of the system's private key, they are eligible to sign the account information for payers and payees. During a transaction one of the trusted nodes will be requested by a payer/payee to sign the account information together based on a threshold signature scheme [11]. After that, it sends the account information to each selected trusted nodes and gets the returned partially signed result, and then it combines all results to acquire complete signed account information, which will be sent back to the payer/payee.

Account Holders are also a quorum of nodes that keep updated account information and records of unspent scrip for a node. *AH* may be selected according to a fixed algorithm, such as the one that chooses one's closest nodes as its *AH* in the Pastry ring [14].

In this approach the consumers will not pay tokens directly for the service, they will use scrip that is issued by the provider and can be used only to get services from the same provider.

2.4 Transferable Debt Token

In the *Transferable Debt Token* scheme [16] a debt token indicates that the holder of the debt token is responsible for redeeming the debt. A peer called *debt token owner* creates a *debt token*, and when another peer purchase from him, he issue the token to the peer, and in the same manner the token can be transferred across peers for payment. Finally, the last token holder has to pay back the money to the owner.

In this scheme each peer P has his own public key pair (pk_P, sk_P), and in the same way, the broker has the public key pair (pk_B, sk_B). In order to ensure security, the messages are signed using the function $Sig_E(m)$; where m represents the message signed under the party $E's$ private key sk_E.

This scheme consists of three basic processes: Registration, Payment and Redeeming.

A. Overview

1. *Registration:* In a first process each peer has to register his personal information (e.g., name, address and phone number) and an account with an initial fixed amount of money in advance. In case the pair has any misconduct, the broker will charge a penalty to the pair's account.
2. *Payment:* Initially a vendor that we will call P_0 creates a *debt token* $DToken_{P_0}$, which indicates that the holder of the token should redeem the token to the owner P_0 before expiration. When a peer P_i purchases services from P_0, P_0 sends $DToken_{P_0}$ to P_i, P_i sends a commitment to P_0 who then provides its service, and P_i becomes current holder of the debt token.
3. *Redeeming:* The last holder of the $DToken_{P_0}$ that we call P_n has to redeem the token to the broker. To this purpose, it should send to the broker a *redeeming request*. After verification, the broker credits money from ${P_n}'s$ account to ${P_0}'s$, and then returns a redeeming proof to P_n. Finally, the broker sends a notice to P_0.

If the debt token has expired and no peer redeems it, the debt token owner asks the broker to identify the last holder of the debt token; he will be identified as a dishonest peer and punished by the broker.

Note that the broker should store the redeeming request until it is expired. If the broker receives two redeeming requests with the same debt token, the broker refuses the second request because the token owner should only issue one debt token for one transaction.

3. ALGORITHM ANALISYS

All or most micropayment for P2P networks schemes proposed to date, agree on some of the key requirements that such schemes should meet, these are:

- Scalability. The workload of any of the network participants should not grow so that it becomes unmanageable (this implies the absence of a central server that participates in all transactions), but rather this workload can be distributed among peers.
- Transferability. Those who received a payment for a service provided, they should be able to use that payment to buy services from any third party without the need of identify themselves with a central broker or currency issuer; and who bought coins to the broker, they can spend their coins on different providers.
- Security. Security mechanisms should seek to ensure that the value of the coins cannot be altered and also prevent double spending.
- Anonymity. As in the traditional commercial transactions, none of the parties involved in a transaction must have need to disclose their identity to any third party and, if possible, also among themselves. To punish misbehavior peers, a reliable judge could reveal the identity of a participant if necessary.

Compliance to these requirements is what we have taken as criteria for analysing each micropayment scheme proposed and that we have considered for this review.

3.1 Scalability in P2P Micropayment Systems

As we have mentioned above, in order to take advantage of the scalability of the peer-to-peer networks, micropayment schemes seek to efficiently distribute the workload generated by transactions among peers, as well as the workload generated for audit each transaction in order to ensure that no fraud can occur in the system. To achieve this goal, three types of implementations are proposed by these schemes:

1. The solutions use a central server or broker but ensure that most of the workload generated by supervisory functions of transactions and security checks are carried out by any of the peers in the network, so the services provided by the broker can be off-line.
2. A broker server is used and also a specific group of peers (generally the most reliable peers or who have

better resources) is chosen to take charge of the coin verification functions and other necessary tasks for transactions.
3. No broker is involved in the verification of transactions but all these tasks are entrusted to a specific group of peers.

CPay [10] exploits the heterogeneity of peers. In this manner, for each transaction a customer peer compute a *consistent hashing function* in order to find a *Broker Assistant* BA to check the coin and decide whether to authorize the transaction. Thus, the Broker is only responsible for selling coins and paying peers as well as the management of BA. The broker is not involved in any transaction only the payer (buyer), payee (seller) and the BA are involved. In this case the BA is responsible for checking the coin and authorize the transaction.

In P2P-Netpay, scheme proposed by Dai et al. [2], Broker is involved in purchasing and redeeming e-coins as well as in verify touchstone when requester first contacts a new supplier, while the process of verifying the validity of the coins and checking the status of the buyer's account is responsibility of the supplier.

The proposal made by Huang and Zhao [9] defines a fully decentralized scheme where no broker is present. In this case two sets of peers called *Account Holders and Trusted Nodes* are defined, here the payee is responsible for double spending and signature verification with the help of the Account Holder. Additionally a Central Registry Server responsible for admitting peers in the system is present.

On the other hand Yen et al. [16] propose a schema with a different approach, in this schema debt tokens are exchanged instead of coins on each transaction, it makes verification mechanisms much lighter and also they can be implemented by the peers involved in the transaction. A broker is also present and is responsible for peer registration management and trace malicious peers in case of fraud.

3.2 Transferability in P2P Micropayment Systems

Transferability is a concept that in this case is also linked to scalability and anonymity, because if there is no need of a central server intervention in the payment process (transfer of exchange), the identity of the parties involved in a transaction will not be compromised with a third party and improve scalability by eliminating the intervention of a single element in all transactions. However its implementation is a trade-off with security, so as we well see in some cases a third party is considered to intervene in each transaction.

CPay [10] uses the concept of transferable coins introduced by PPay [15], and sheds the checking work unto the Broker Assistants, which are a set of peers that have more computational resources and credibility. In CPay, money can be transferred in two ways:

1. As Coins where the broker's signature is present (when a peer first bought coins from the broker).
2. As Authorization Message, representing that BA authorizes a given pair to pay to its provider using a specific coin.

P2P-NetPay [2] uses a one-way hash function to generate a *"payword chain"* which represent a set of e-coins. P2P-NetPay supports transferability between peer-vendors without extra actions on the part of the peer-users, here the payword chain is transferable to enable users to spend e-coins in the same e-coin (payword) chain to make number of small payments to multiple vendors. An *"index"* is used to indicate the current spent amount of each e-coin chain and should be verified by the provider before authorize a transaction.

Unlike the previous cases where an e-coin or commitment is transferred so that the last token or e-coin holder can deposit it for money, Yen et al. [16] proposes a new micropayment protocol based on transferable debt token, where the token itself is a *debt*, thus, the last token holder has to redeem this debt. In this approach the token is transferred from the seller to the buyer without the intervention of a third party and the services are provided once the seller has verified the commitment of the buyer.

On the other hand, on the approach proposed by Huang and Zhao [9] transferability of the tokens between peers is no present. Here, a concept of scrip is introduced which is issued by a provider, the scrip is marked with the identity of the provider and the usage, the buyer must purchase scrip from provider so that he can use them to pay for the service specified in the scrip and that was received from the same provider.

3.3 Security in P2P Micropayment Systems

For any of these schemes to succeed, they must have reliable security mechanisms for timely detection of any fraud including double spending. Preventing fraud in the case of micropayment systems for peer-to-peer networks is difficult to achieve without compromising performance and scalability, especially when the broker implements these detection mechanisms. So many security implementations are proposed and described below.

CPay [10] assumes that the broker signature is reliable, and under this assumption argues that, since the broker signature is present in either of the two ways of transfer coins, then it is impossible that the coin is forged or doubly-spend.

As we see above, in P2P-NetPay [3] each coin is part of a set called *payword* or *e-coin chain*, which is generated

Table 1. Micropayment for P2P Networks Algorithm comparison.

	Scalability	Transferability	Security	Anonymity
CPay	*Broker* is only responsible for selling coins and paying peers as well as the management of *BA*. *BAs* are responsible for checking coins and authorize transactions.	Coins are transferable and can be transferred as *Coin C* or *Authorization Message Au.*	Security relies on the reliability of the *broker*'s signature, which is present in either of the two ways of coins transfer.	Not supported in its original version. An extension to the protocol is proposed.
P2P NetPay	*Broker* is only involved in purchasing and redeeming e-coins. Vendors are responsible for verify validity of e-coins and checking the status of the buyer's account	*Touchstone* and *Index* are transferable between vendors. *e-coins* from the same payword chain can be used to pay multiple vendors.	A *Touchstone* generated and signed by the *broker* is used to verify the authenticity and integrity of an e-coin. An *Index* is used to prevent double spending.	A pseudonym ID_C is used to ensure anonymity of the peers in transactions.
A Secure and Lightweight Micropayment Scheme in P2P Networks	Fully decentralized scheme. Broker or Central Registry Server is responsible for admitting peers in the system. A set of *Trusted Nodes* and *Account Holder* are used to manage the system's signature and the peer's accounts. Vendor is responsible for double spending and signature verification.	No transferability is present; transactions are carried out in a single vendor client relationship. Scrip generated by the provider are used to pay for a service	Trusted nodes use threshold signature scheme [11] to sign peer's account information. Provider signs the scrip using a secret key and one hash operation.	No anonymity is present in this scheme.
Transferable Debt Token	Broker is only responsible for peer registration, management; and trace malicious peers in case of fraud. The peers involved in the transactions implement verification mechanisms.	Debt Token is fully transferable from seller to buyers.	Each peer and the broker has theirown public key pair in order to ensure security	Only the identity of the provider is revealed in the debt token.

from a *root element* using one-way hash function. This root element is named *"touchstone"* and it is transferred between each provider that has been paid with e-coins from the same e-coin chain so they can verify the validity of the e-coin, thereby implementing a low-cost but secure method. Additionally, an *index* that is transferred along with the touchstone is used to indicate the amount of e-coin spent, which prevents peer users from double-spending and peer vendor from over debiting.

Recall now that in Huang and Zhao proposal [9] scrip are used to pay for a service provided, scrip is generated by a provider and only can be used to pay services received from this provider; then, in order to implement security the scrip is signed by the provider, this sign is generated using a secret key (chosen by the provider) and one hash operation, and as it is the same provider that generates the signature which must verify, there is no need to share the key.

As in Yen et al. proposal [16] a debt token is transferred, authors show that any attempt of fraud would de unprofitable. However each peer in the scheme has its own public private key pair in order to implement security.

3.4 Anonymity in P2P Micropayment Systems

Traditional payment methods for online transactions like credit cards or online payment systems have flaws of privacy for the parties involved in a transaction, since the identity of the buyer and seller is exposed not only between them, its also disclosed to the financial institution that manages the transaction, this can reveal precious or sensitive information about the parties involved. That is why actual proposals are seeking payment mechanisms and in this particular case micropayment mechanisms implement anonymity as in traditional commercial transactions.

In its original proposal CPay [10] not support anonymity; this is the reason that the authors propose *Anonymous CPay* as a variation to the original protocol, which offer anonymity by the use of encryption, so the *Broker Assistant* will not know who the payee is.

In P2P-NetPay after pairs are registered with the broker, each one of them receives a pseudonym (IDc), which is used as identity during transactions. Since only the broker knows the mapping between pseudonyms and the true identity of the peers, the privacy is guaranteed.

Meanwhile, Huang and Zhao proposal [9] focuses on provide a lightweight protocol but sure, so anonymity mechanisms are not specified; the identity of the buyer and seller are present in each scrip and are verified on every transaction, but as each scrip is used for a specific transaction, the parties involved only manipulate it. Additionally, in the event that the scrip is stolen, the anonymity of the parties is assured since the provider signs it with its secret key, which is not shared with anyone.

Also, in Yen et al. proposal [16] no specific mechanism for anonymity is present, however from the protocol we can see that only the identity of the provider is present in the

debt token created by it, and that is transferred from peer to peer.

Thus, we can find in the schemes many similarities as well as specific ways to implement the same concept, which are summarized in Table 1.

4. CONCLUSIONS

Micropayment systems presented to date generally follow similar objectives in trying to implement scalability, security, transferability, and anonymity in their designs, although the implementation of the two latter generates a trade-off between security and performance. With the premise of minimizing the impact that the workload generated by the security mechanisms may cause in the system, mechanisms with "good enough" security are procured while the workload of the broker is reduced and distributed among the peers. These micropayment systems also seek to minimize the costs of security mechanisms, as these would exceed the cost of the transaction itself. Although each of the proposed mechanisms analyze their performance in different environments, it is necessary to evaluate all them in a same environment in order to be more objective in their assessment. That is why we propose as future work the use of Agent Based Modeling structures to analyze how these proposals improve the P2P network performance.

REFERENCES

[1] E. Adar y B. A. Huberman, «Free riding on Gnutella», 02-oct-2000.

[2] K. Chaudhary y X. Dai, «P2P-NetPay: An Off-line Micro-payment System for Content Sharing in P2P-Networks», J. Emerg. Technol. Web Intell., vol. 1, n.o 1, ago. 2009

[3] X. Dai y J. Grundy, «Off-Line Micro-payment System for Content Sharing in P2P Networks», en Distributed Computing and Internet Technology, vol. 3816, G. Chakraborty, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 297-307.

[4] X. Dai y J. Grundy, «NetPay: An off-line, decentralized micro-payment system for thin-client applications», Electron Commer Rec Appl, vol. 6, n.o 1, pp. 91–101, ene. 2007.

[5] X. Dai y B. Lo, «NetPay – An Efficient Protocol for Micro-payments on the WWW». Fifth Australian World Wide Web Conference, Australia, 1999.

[6] S. Glassman y M. Manasse, «The MilliCent Protocol for Inexpensive Electronic Commerce».

[7] P. Golle, K. Leyton-Brown, I. Mironov, y M. Lillibridge, «Incentives for Sharing in Peer-to-Peer Networks», en Electronic Commerce, vol. 2232, L. Fiege, G. Mühl, y U. Wilhelm, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 75-87.

[8] R. Hauser, M. Steiner, y M. Waidner, «Micro-Payments based on iKP», presentado en Proceedings of 14th Worldwide Congress on Computer and Communications Security Protection, Paris-La Defense, 1996, pp. 67-82.

[9] Q. Huang y Y. Zhao, «A Secure and Lightweight Micro-Payment Scheme in P2P Networks», en International Conference on Industrial and Information Systems, 2009. IIS '09, 2009, pp. 134 -137.

[10] Z. Jia, S. Tiange, H. Liansheng, y D. Yiqi, «A new micro-payment protocol based on P2P networks», en IEEE International Conference on e-Business Engineering, 2005. ICEBE 2005, 2005, pp. 449 -455.

[11] C. Park y K. Kurosawa, «New EIGamal Type Threshold Digital Signature Scheme», Ieice Trans. Fundam. Electron. Commun. Comput. Sci., vol. E79-A, n.o 1, pp. 86-93, ene. 1996.

[12] R. L. Rivest, «The MD5 Message-Digest Algorithm». RCF1321, abr-1992.

[13] R. L. Rivest y A. Shamir, «PayWord and MicroMint: Two Simple Micropayment Schemes», presentado en Proceedings of the International Workshop on Security Protocols, 1996, pp. 69-87.

[14] A. I. T. Rowstron y P. Druschel, «Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems», en Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, London, UK, UK, 2001, pp.

[15] B. Yang y H. Garcia-Molina, «PPay: micropayments for peer-to-peer systems», en Proceedings of the 10th ACM conference on Computer and communications security, New York, NY, USA, 2003, pp. 300–310.

[16] S.-M. Yen, K.-Z. Chiou, J. Zhang, y P.-H. Lee, «A New Peer-to-Peer Micropayment Protocol Based on Transferable Debt Token», en Transactions on Computational Science X, vol. 6340, M. L. Gavrilova, C. J. K. Tan, y E. D. Moreno, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 352-363.