A Secure P2P Incentive Mechanism for Mobile Devices

Jani Suomalainen[1], Anssi Pehrsson[2] and Jukka K. Nurminen[3,4]
[1]VTT Technical Research Centre of Finland
P.O. Box 1000, FI-02044 VTT, Espoo, Finland
Jani.Suomalainen@vtt.fi
[2]Small Planet Ltd
Ruoholahdenkatu 8, FIN-00180 Helsinki, Finland
Anssi.Pehrsson@smallplanet.fi
[3]Nokia Research Center
P.O. Box 407, FI-00045 Nokia Group, Helsinki, Finland
Jukka.K.Nurminen@nokia.com
[4]Helsinki University of Technology
P.O. Box 5400, FI-02015 TKK, Espoo, Finland

Abstract

Peer-to-peer applications are emerging into mobile devices. However, resource limitations of these devices introduce new challenges for P2P technologies. For instance, there is a need for incentive mechanisms, which address the free riding problem but do not waste devices' battery or communication resources. A centralized and user-identity based incentive mechanism enables mobile users to contribute with any device and receive P2P services with mobile devices. We explore security issues related to a centralized incentive mechanism by analyzing and classifying threats and potential security mechanisms. We propose a privacy preserving security architecture. The architecture is based on authentication, software tamper protection, and misbehavior detection mechanisms. Further, we describe a prototype implementation for mobile BitTorrent file sharing peers. We provide a discussion on potential security compromises, not jeopardizing sufficient security level, and compare our work to related research.

Keywords: *P2P; security analysis; incentive; mobile devices; BitTorrent*

1. Introduction

Peer-to-peer (P2P) based applications, particularly content sharing, are currently popular in personal computers and are expected to gain popularity also in mobile devices. Mobile devices have already enough computing and communication capabilities enabling them to participate to P2P networks. However, there are still challenges, which hinder mobile phones participation and contribution. For instance, incentives for users with mobile devices to contribute are not clear. Energy efficiency and communication costs are critical issues, which differentiate mobile devices from personal computers and discourage mobile users' contribution. Consequently, there is a need for incentive mechanisms, which motivate mobile users to contribute but also save mobile device's limited resources.

To address the problem of free riding, users can be rewarded for their contribution to the network. Different incentive mechanisms have been proposed and adopted to P2P networks. Approaches include distributed schemes, where either the contributor itself (P2P client software) or peers monitor contribution, and centralized schemes, where servers maintain records on clients' contribution. In Section 6, we present a survey on existing work related to P2P incentive mechanisms. However, existing schemes have not been designed from the point of view of mobile devices. Hence, we have made an own proposal for an incentive mechanism: the credit system. This mechanism ties rewards to user-identities instead of device-identities and is thus more suitable for users with different kinds of terminals including mobile devices.

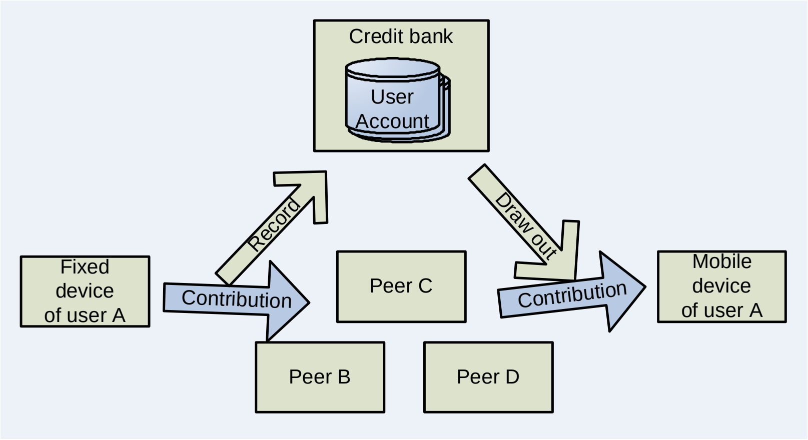

Figure 1. An overview of the P2P credit system. The user A contributes to P2P network with a fixed device, without resource limitations and consumes credits using mobile device, with limited communication and energy resources. The credit bank keeps track of users' contributions and controls how much contribution the user may receive from other peers.

Unfortunately, as the credits used in the incentive mechanism can be monetarily valuable, this system will face different threats from attackers trying to misuse the system. Therefore, security mechanisms, which are feasible also for mobile devices, are needed.

In this paper, we study the credit system from the point of view of security. The previous version of the paper [1] was presented in the ICIW 2008 conference. This version has been extended with more extensive security analysis and literature survey. First, in Section 2, we review our proposal, the credit system. The system was initially proposed and its feasibility evaluated through a mathematical analysis in [2]. In Section 3, we describe threats and potential attacks against incentive mechanisms and the system. In Section 4, we describe security architecture for the credit system. We contribute by surveying and analyzing which security mechanisms are available and how they could be applied for the credit system. In Section 5, we describe a prototype implementation of the credit system for BitTorrent clients, which are running on mobile devices. In Section 6, we compare of existing incentive mechanisms against the proposed solution.

2. The credit system

The proposed P2P credit system is a centralized incentive mechanism, which enables mobile devices to participate P2P network without requiring mobile terminal itself to contribute. Architecture of the credit system is illustrated in Figure 1.

The central entity of the system is the credit bank, which rewards P2P nodes for their contributions with credits and controls that only those nodes with credits can receive services from the network.

Credits are user-specific as the credit bank maintains accounts for each user. This enables the same user to collect credits with different devices. Also, credits can be used with any device belonging to the user or given for other users. This enables mobile users to receive services from P2P networks even if they do not want to contribute with their mobile terminals. For instance, a user can contribute with a PC at home and then use credits from this contribution with a mobile device.

Contribution, providing credits, may mean e.g. sharing content, supplying information on content location, or performing computations. Different contributions may be valuated differently. For instance, sharing of DRM protected content may provide more credits than sharing of unidentified data. Credits can be utilized to get content, services or high quality of service (QoS) level from other peers or, alternatively, from external service providers.

Credits, which can be utilized in other devices, provide incentive for high-capacity servers i.e. 'super-nodes' to contribute. This would motivate commercial service providers to participate P2P network. However, commercial parties require strong security measurements.

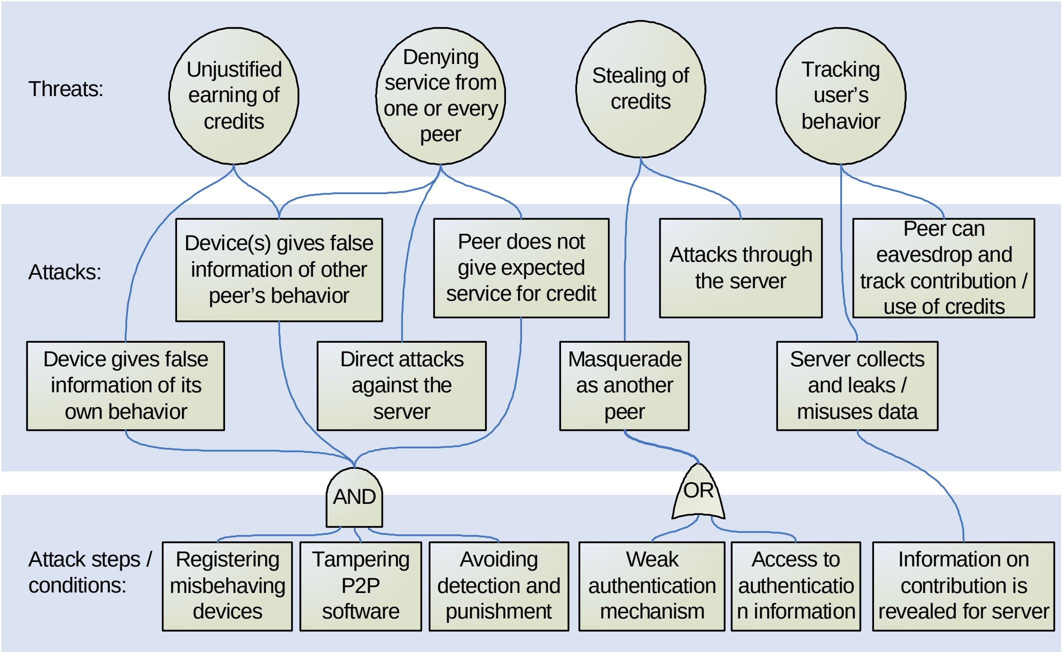
Threats:
Unjustified earning of credits
Denying service from one or every peer
Stealing of credits
Tracking user's behavior
Attacks:
Device(s) gives false information of other peer's behavior
Peer does not give expected service for credit
Attacks through the server
Peer can eavesdrop and track contribution / use of credits
Device gives false information of its own behavior
Direct attacks against the server
Masquerade as another peer
Server collects and leaks / misuses data
AND
OR
Attack steps / conditions:
Registering misbehaving devices
Tampering P2P software
Avoiding detection and punishment
Weak authentication mechanism
Access to authenticatio n information
Information on contribution is revealed for server

Figure 2. Threats against P2P credit systems. Threats have been classified to four main categories. The image also illustrates some attacks and related attack phases, which may make these threats true.

3. Security threats

Participants of the credit system, the credit bank and peers, may be attacked in different times. Attacks may occur during the contribution phase, after contribution, or in consumption phase. Some examples of security threats are listed in Figure 2. The figure provides also an attack tree illustrating some attacks, which might realize these threats.

The contribution phase is vulnerable for various attacks, where an attacker tries to earn unjustified credits:

1. A device may claim that it contributed, even if it did not, in order to receive credits
2. Nodes may give false (positive) information about their peers. For instance, a user may have two devices giving false information of each other. Also, in the 'Sybil attack' [3], an attacker may have a large amount of virtual peers providing false information. Further, different users may also collaborate and e.g. exaggerate each others' contribution.
3. A device might contribute but the contribution may be bogus. For instance, a device may claim that it made particular analysis of given data without doing it or uploaded content files may be corrupted.

In order to execute attacks, which require peer to give false (positive or negative) information, an attacker must have suitable attack software. This can be achieved by tampering authentic software. Tampering attacks require some skill but after tampering the attacker may distribute attack software to other users through Internet.

The credit system may face different availability related threats:

1. As the credit system is dependent on a centralized credit bank server, it is vulnerable for denial-of-service attacks. These attacks may utilize protocol vulnerabilities in the credit bank server or be brute-force attacks.
2. Devices may give false information about their peers and claim e.g. that peer's contribution was not acceptable. As a consequence, the credit bank may limit victim peer's access to its credits.
3. A peer may claim that a user received contribution in order to decrease amount of user's credentials. This attack may occur when user is expecting service or, potentially, at any time when there are credits in users account.

A credit bank or communication between peers may be attacked in order to steal credits or to get services with credits belonging to others.

1. An attacker may tamper identity information of contribution made by others. This may be possible if peers are not authenticated or if authentication mechanisms have security vulnerabilities.

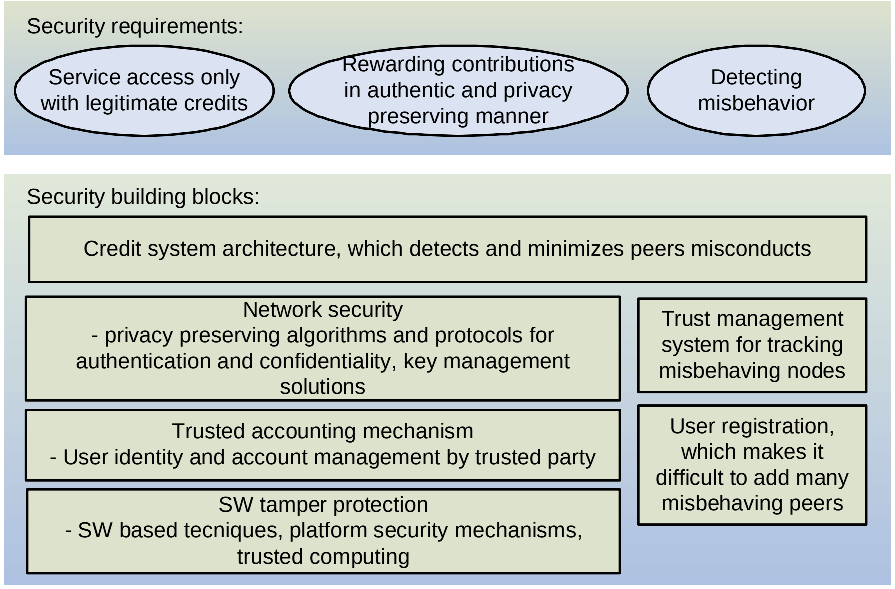
Security requirements:
Service access only with legitimate credits
Rewarding contributions in authentic and privacy preserving manner
Detecting misbehavior
Security building blocks:
Credit system architecture, which detects and minimizes peers misconducts
Network security
- privacy preserving algorithms and protocols for authentication and confidentiality, key management solutions
Trust management system for tracking misbehaving nodes
Trusted accounting mechanism
- User identity and account management by trusted party
User registration, which makes it difficult to add many misbehaving peers
SW tamper protection
- SW based tecniques, platform security mechanisms, trusted computing

Figure 3. Security requirements and building blocks. The credit system's security architecture is designed to fulfill the three main requirements: Services can be accessed only with legitimate credits. All contributions are rewarded in authentic and privacy preserving manner. Misbehaviors may be detected even if attacks may occur. These requirements are addressed with the underlying security building blocks.

Alternatively, attacker may gain access to authentication information e.g. through malicious software, which is installed to user's device.

2. An attacker may utilize some vulnerability in the credit bank server and gain an access to the server. Then attacker may e.g. modify credit databases.

Peers' activities in P2P system may be tracked and using this information the user may be profiled. For instance, a particular fear is that users may be punished due to their contribution. A centralized credit system provides some new security worries, which should be considered. Firstly, the credit bank provides a single point, which must be trusted and which can be monitored or attacked to resolve information. Secondly, participation with different devices can be mapped to a single user.

4. Security building blocks

A design of a secure P2P credit system must consider the threats identified in the previous section. Essentially, the system must control that service are available only if credits are used in legitimate manner and that rewards for contributions are given for authentic contributors. Further, it is preferable that the system does not cause any new privacy problems. This control can be proactive. However, in practice, proactive solutions cannot provide full protection, and hence there must be a way to detect misbehaving peers. Secondarily, as scalability is a potential bottleneck of the server based credit system, one goal for security architecture should be minimization of required communication and computing resources. Figure 3 illustrates the main requirements for the system and discussed security building blocks. In the following subsection, the secure credit system architecture is first described. Then, design decisions and available building blocks are further discussed.

4.1. Architecture

In the P2P credit system architecture, both contributor and consumer inform the credit bank when a contribution is made. Two sources are required, as individual nodes cannot be trusted to deliver correct information. A message diagram in Figure 4 illustrates communication in the secure P2P credit system. The figure illustrates a basic case where User B contributes with one device and User A consumes credits with another device.

The consumer initiates scenario by requesting contribution from a contributor and by authenticating itself. The contributor will check from the credit bank if the consumer has enough credits for the requested amount of contribution. If there are enough credits, the contribution begins. The credit bank increments User B's account after the credit query has been made.

When the consumer has received the contribution, it will inform the credit bank, which will remove credits from User A's account. Alternatively, the credit bank could remove credits already, when the contributor makes the query. The latter approach would save some signaling costs but is infeasible since it might cause users to loose credits when a contributor goes offline or crashes due to technical failure. The consumer might try to get contribution for free by not sending a verification message. However, the credit bank is able to detect peers who make large amount of content queries but do not make any contribution verifications.

To make the system more scalable a few mechanisms can be applied to minimize amount of communication.

1. A buffering mechanism can be utilized to avoid messaging between peers and the server during every transaction. Contributors and consumers can buffer information of transactions and send larger reports only occasionally e.g. once per a day. After noticing that account balance has gone to negative, the credit bank will block user's participation by not renewing user's authentication information (e.g. time limited certificates).
2. A contributor does not need to inform the server on every transaction. For instance, to save battery resources, a mobile node may choose to not to make confirmations. The consumer should not be able to determine whether a confirmation is made or not and, hence, should not be able to send verification messages at the same time.
3. Some resource optimization can be achieved by selecting which contributions are rewarded and which require credits. For example, credits can be demanded only from information of locations content files instead of demanding them for every small part of content file.

4.2. Authentication mechanisms

Network security mechanisms – security algorithms and protocols – are needed to authenticate communication. The strength of an authentication mechanism should be selected so that efforts of attacking are larger than efforts of contributing. If contribution means uploading of files, cryptographic authentication of contributor may not be needed. This is because capturing, tampering and then uploading a tampered file of may be more difficult than uploading own files. However, if contribution means running some program for some period of time before transmitting, stronger authentication is required.

When there is a large amount of messages between peers and the credit bank related to small contributions, it may not be justifiable to make too heavy and resource consuming authentication. Authentication protocols based on shared secrets may be more feasible, instead of protocols utilizing asymmetric cryptography or heavy handshakes such as TLS.

Single sing-on architectures, for instance solutions from Liberty alliance and Microsoft passport, provide potential authentication infrastructures, which could be adopted also for the credit system. These systems phase similar challenges i.e. enable nodes to authenticate themselves to different servers (in our case other peers and the credit bank).

4.3. Software tamper protection

Peers and P2P software in them cannot be assumed to be trustworthy. A single attacker may modify one copy of the client software and then distribute this tampered version to other users. However, with software and device security mechanisms some additional trustworthiness may be gained.

Some security level can be achieved with obfuscation techniques, which make changing program code more laborious and time consuming. However, determined attackers can circumvent obfuscation based security.

Another approach is to use trusted hardware modules. For instance, trusted computing technologies enable small trusted hardware components to verify identity and integrity of software running in a device. Consequently, the credit bank or contributors could remotely attest and verify that a client device is running authentic software. These remote attestation mechanisms have been proposed also for P2P environments [4]. However, efficiency and scalability issues may limit the usability of remote attestation. Also, current platform security mechanisms in mainstream mobile devices do not support these mechanisms.

4.4. Detecting misbehavior

When every device cannot be assumed to be trustworthy, mechanisms for detecting misbehaving peers are needed. Particularly, there must be a way to monitor and analyze suspicious actions and there must be a way to punish misbehaving clients.

Clearly, suspicious activities for the credit system include cases where a peer makes credit query but a consumer does not confirm to receive content. In individual cases, one suspicious activity is not an evidence of misbehavior or does not indicate who the faulty counterpart is. However, a large amount of suspicious activities might indicate illegitimate behavior.

Detecting misbehavior becomes more challenging if attackers are able to easily introduce large amount of (virtual) misbehaving nodes or to change identities when the credit system tries to punish the user. In order to be able to defend against these attacks, the registration process should not be too easy or cheap. At least, an attacker should not be able to automatically add new virtual nodes.

One characteristic of attacks trough virtual nodes is that these nodes will get most of their credits from the same peers. Therefore, these attacks might be detectable by looking for isolated groups where some peers get exceptionally many contribution verifications. Unfortunately, this kind of mechanisms would detect also users whose contribution is interesting only for some very specialized users. Hence, this kind of mechanism would be an incentive for users to contribute content that is popular for masses. Also, analyzing this kind of behavior would probably be unfeasible for large amount of peers.

Attacks where registered peers collaborate are difficult to prevent. Active manual work may be used against some attacks. For instance, tampered software, which multiplies the amount of notified contribution, may be detectable when it communicates with other peers (these peers must agree on the amount of informed contribution). If these clients emerge, detectors must implement new mechanisms for tracking misbehaving clients.

Punishment mechanisms depend on the nature of P2P network and value of content. In minor cases, available service level could be cut down for potentially suspicious devices. For instance, an account can be decremented or frozen for some period of time. When the monetary value of credits is significant, judicial actions could be possible.

4.5. Privacy enablers

The P2P credit system should not introduce any new unnecessary mechanisms, which would further compromise privacy.

Peers, receiving and verifying contribution, do not themselves need to identify peers who are contributing. However, the credit bank needs to map verifications into contributors. To enable peers to verify contributions without revealing identity to peers, temporary random identifiers can be utilized. Consequently, an attacker cannot utilize these identifiers to determine if different contributions are made by one user and not by several users. It is enough that the credit bank is able to map users' accounts into random identifier. This mapping can be enabled with a message exchange where peers request temporary identifiers from the credit bank. However, message exchange for every contribution means additional communication. A better solution might be that contributors and the credit bank agree shared secrets, which they use to generate identifiers. For example, pseudo random sequences [5] could potentially be applied in a P2P credit system.

As a consequence, use of random identifiers enables a credit system to work with anonymous P2P networks, such as Tarzan [6] and Freenet [7]. Use of temporary identifiers prevents also attackers, who are eavesdropping communication between peers and the credit bank, from resolving peers' communication parties. Alternatively, cryptographic solutions could be utilized to achieve the same effect. To prevent eavesdropper from resolving how much peer is contributing, encrypted bogus traffic could be introduced. However, for mobile devices use of cryptographic techniques and bogus traffic is expensive.

The credit bank needs information on contributors' identity as well as the amount of contribution. Also, the credit bank may require information of real identities in order to implement strong misbehavior detection system. However, information on exact contribution does not have to be revealed.

5. An implementation of the credit system for mobile BitTorrent clients

To evaluate the feasibility of the credit system idea, a prototype was implemented and requirements for security enhancements studied. This prototype contained a mobile application for the BitTorrent-file sharing protocol [8] and a centralized credit bank implementation. The prototype also contained a BitTorrent tracker that stores information of shared files and their locations.

The mobile peer application was implemented with Java Micro Edition (Java ME). The credit server, which communicated with peers over HTTP protocol, was implemented with J2EE Servlet Technology. For persistency each credit transaction was written to a RDBMS, which also contained user credentials. Passwords were stored to the database in MD5-hashed form to ensure password security inside the server. Apache Tomcat 5.5 was used as a Servlet runner and MySQL 4.1 as a database server. Mobile application was implemented using MIDP 2.0-standard with JSR 75 extension, providing capabilities to read and store files. This peer application was developed and tested with Nokia E65 having Symbian 9.1 operating system.

The credit system introduces additional messaging for BitTorrent clients. Each time a certain file piece was uploaded or downloaded by peer, the credit server is informed of the transaction. This credit server communication was implemented by sending messages in a BitTorrent-specific bEncoded form over HTTP. This way existing logic for data structure handling in peer applications could be reused. These actions received by the credit server were then written to database, and appropriate accounts were compensated respectively.

The implementation works with existing peer applications without changes to the torrent protocol. Figure 4 shows these existing communication sockets with dashed lines. In addition to these sockets, each peer application communicates with credit server with separate connections. These connections are drawn with solid lines in Figure 4. Using separate sockets for additional credit communication allows peers which have not been integrated to credit system to use existing torrent network without problems. The model also enables torrent tracker to communicate with the credit server. This makes it possible for a torrent tracker to prioritize peers while informing others of content availability. This decision could be based on contributing peers' credit balance and contribution actions.

The credit system requires that communicating peers are able to identify and authenticate each others. The BitTorrent protocol introduces a peer identifier to identify peers from each other, but there is currently no logic to ensure that this identifier is globally unique. Currently peer identifier allocation depends on the BitTorrent client implementation, and several of implementations even use all random numbers while generating this 20-bytes long identifier. In current versions of the protocol there is no structured way for constructing such identifier, although some conventions have been applied in Azureus and Shadow's-styles.

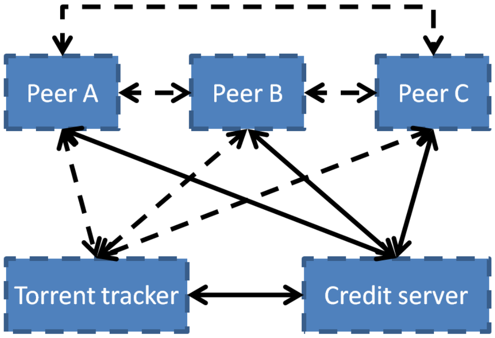

Figure 4. Communication sockets in the prototype. Credit specific communication happens through separate sockets (solid lines). The Prototype does not change existing BitTorrent protocols (dashed lines). The approach enables cooperation with legacy BitTorrent peers.

For the credit system, the peer identifier should not only be unique, but the credit bank should also be able to persistently identify peers between sessions as well. This way the credit server is able to remember contributions performed by certain user between different sessions and terminals. The prototype implementation used user credentials to identify each peer; the credit server user name and password could be changed from BitTorrent peer user interface. Each earned and consumed credit is then handled from a credit account mapped to credit server user name. The communication is protected with a password to prevent non-authorized peers to consume credits earned by someone else. When credit allocation is based on credit accounts, users are able to use various different peer applications even at the same time to earn and use credits.

In BitTorrent, there is not a way to deliver peerid to other peers in trustworthy manner. As discussed in Subsection 4.2, authentication of contributors does not have to be very strong. The costs of attacks are likely to be larger than the costs of contributions. Consumers, however, should be authenticated, so that peers are not able to use credits belonging to other users. Identifiers could be protected with cryptographic methods based e.g. on certificates or shared secrets. This would require either a change in the BitTorrent protocol or an implementation of a new transfer mechanism, in order to authenticate BitTorrent peers. Implementing another socket between the peers is not that reasonable architecture – mobile terminal would need additional server socket and twice as much transfer sockets when compared to existing protocol. Implementing such feature would also increase resource consumption in the terminal. At the same time, modifying the existing protocol would strict the usage of the peers for specific BitTorrent implementations only.

The prototype does not provide strong peer authentication. Instead of modifying the existing BitTorrent-protocol, the prototype server kept track of peerids, mapped with peers' public IP-addresses. This way the torrent tracker and the credit server always knew the peerid in each tracker request, even when BitTorrent client applications did not explicitly define them. The use of IP addresses brings some protection against attacks where consumer fakes its peerid in order to use other users' credits. These attacks are complex as they must include the network infrastructure so that the files are routed to the attacker instead of the correct owner of the IP address. This approach is relatively simple and requires public and unique IP-address for each peer, which is not currently the case in the real world.

The forthcoming implementations may utilize identifiers, which are allocated by the credit bank. Central allocation of temporary and random user-specific identifiers would also enable privacy as discussed in Subsection 4.5. Also, it is possible to utilize security protocols such as TLS for authenticating consumers and for authenticating peers' communication with the credit bank.

6. Related work

Existing efforts for P2P incentive mechanisms can be classified into three broad approaches: some devices monitor how peers behave, some devices have trusted client software to monitor user behavior, and some devices rely on other peers to monitor how peers behave. Additionally, there are incentive efforts with wireless devices utilizing either trusted hardware or cooperative trust management schemes.

6.1. Monitoring BitTorrent peers

BitTorrent clients already now implement choking or tit-for-tat algorithms [9], which provide an incentive for peers to contribute. The purpose of this mechanism is to enable individual peers to maximize own download rates by selecting best peers. BitTorrent does this by monitoring how much peers contribute and then choking, i.e. temporarily refusing to upload for, those peers providing the worst service.

Each peer selects a fixed amount for peers to be choked once every ten seconds. Peers selected for choking are in principle those, which provide the worst download rate. There is also an optimistic unchoke algorithm. In optimistic unchoke, peers, which have previously provided bad download rate, are given change to provide better performance during a thirty seconds period. If the performance improves over that period, the unchoking of peer is continued. There is also a so called anti-snubbing mechanism. When a client does not receive any data from a peer for one minute, it assumes that that peer has choked it and stops uploading to that peer except during optimistic unchokes.

This tit-for-tat incentive mechanism is essentially file-specific. It provides incentive for peers to share a file at the time they are downloading the file. The algorithm is trustworthy as download rates are measured in the peer, which is also rewarding peers for contribution by uploading content.

When comparing BitTorrent's tit-for-tat mechanism to our credit system, we can see some differences.

Firstly, tit-for-tat does not enable users to upload at different time and download at another time or with another (mobile) device. This means that non-contributing mobile peers will get bad service when tit-for-that model is used. Whereas, the credit system we proposed and implemented for BitTorrent enables collecting rewards at the later time and with different devices. Hence, our model is more suitable for encouraging long term good behavior.

Secondly, BitTorrent's tit-for-tat mechanism is symmetric where a peer rewards only those peers it is communicating directly. In many cases P2P connections are asymmetric. For instance, a peer may be a single contributor of a rare file but may utilize several sources to download a popular file. The proposed credit system is fairer as contributions are evaluated from a point of view of P2P community instead of an individual peer.

Thirdly, tit-for-tat mechanism is vulnerable for selfish peers. For instance, a modified BitTorrent client, BitTyrant [10], showed that peers can receive more that they contribute by carefully selecting contributed peers and upload rates. Also, as noted in [11] the punishment comes within delay and the peers may easily change identities since BitTorrent does not provide strong authentication mechanisms. Our proposal addresses these threats by keeping track of contribution for longer time period and by being compatible with stronger authentication means.

On the other hand, BitTorrent's tit-for-tat mechanism is more efficient mechanism for selecting optimal peers to communicate with when uploading a particular file.

6.2. Local contribution tracking

Some existing P2P technologies, such as KaZaA, have incorporated own credit mechanisms to client software. These mechanisms track how much the end-user uploads and, according to that information, locally adjust download rates.

These solutions are efficient, easy to implement, and scalable, as they do not require contribution from peers or servers. Also, they keep track of contributions for longer time scale than e.g. BitTorrent's tit-for-tat and thus will provide incentive to contribute also when the user is not downloading.

However, these mechanisms are tied to particular devices and do not consider resource limitations of mobile devices. Also, these local solutions are vulnerable for tampering attacks. For instance, KaZaA Windows clients' participation level information has been stored on Windows registry in obscure format. Users have been able to modify information according to easy guidelines, which are available on the web sites such as [12]. Architectures where trust is not tied to consumer side software, including our proposal, are less vulnerable for tampering.

6.3. Distributed incentive systems

Some research proposals have adopted remote incentive schemes where either a centralized server, as our credit bank, or other peers are used to track contribution and to control which nodes can be provided rewards. Two basic types of approaches for storing and protecting accounted contribution have emerged:

Remote accounts – In these proposals information on contributions is tracked into an account, which is stored in a centralized or distributed repository. The account management is done by a trusted party.

Cryptographically protected electronic currency - In these schemes, peers get tokens from contribution and use tokens to receive service. The advantage of these schemes is that they do not require active participation of a server, which might become a bottleneck. Server's participation may be required only for some operations such as for initial registering to the network and for preventing double spending.

BitStore [13] is one approach proposing remote currency based incentive scheme for BitTorrent. It has been originally designed to address BitTorrent's problem that there may not be complete sources available, especially for rare files. BitStore is a P2P network, which keeps complete copies of content and which is parallel to the BitTorrent network. Participating peers are rewarded with tokens, which are cryptographically protected. The value of tokens depends on an auction based market mechanism.

BitStore is similar to our approach in a sense that it uses centralized nodes to control peers' transactions. BitStore uses trackers as trusted third parties for controlling money and token transactions. However, BitStore does not address challenges of mobility nor bind tokens to particular users, which have been done in our proposal.

PPay [14] has adopted a token based scheme. In PPay, reliance on server is kept on minimum as server is not needed in normal transactions. Frauds are made detectable by leaving to tokens an audit trail, which identifies who has used them. Our proposal is different than PPay in a sense that we require more server interactions. However, by doing this we make double spending attacks proactively impossible. Also, we address one problem of intensive mechanisms (a peer who has collected large amount of credits stops contributing altogether) by enabling adjusting of account balances in flexible manner. For instance, credits could be periodically withdrawn from accounts using some algorithm so that any user cannot that completely stop contributing.

PeerMint [15] introduced remote accounting based solution for an incentive mechanism, which is both reliable and scalable. They used an overlay P2P network to keep store users accounts. In PeerMint, both the contributor and the consumer inform accounts of both peers for a transaction that has taken place. The accounts may not locate in the same peer. To make the system more manageable, PeerMint uses session specific mediator peers, which are informed on contributions during sessions and which at the end of sessions then update accounts of participants.

E-cash [16] proposes a token based approach where users can withdraw tokens from an account in a server, called the central bank. After earning tokens, the user must deposit them to the bank before they can be used. This enables the server to track use of coins and to detect frauds. Sending tokens to server after every transaction and then withdrawing them will also cause additional overhead. The paper [16] does not directly address limitations due to mobility. However, the approach is similar with our proposal when the account holders deposits tokens with a fixed terminal and withdraws them with mobile devices.

6.4. Incentives for wireless devices

Some research work has been given for studying incentives in wireless ad hoc networks. These networks rely on peer nodes to voluntarily route others traffic. Since these nodes typically have very limited battery capacities, they typically have no incentive to do this. Incentive mechanism proposed for wireless ad hoc networks are different from P2P environments in a sense that they are smaller and cannot be assumed to have any long-lived central authority.

Incentive mechanisms based on cryptographically protected electronic currency have been proposed for ad hoc networks. For instance, Nugglets, presented in [17], is a token based approach. This approach does not assume that there is any centralized authority, which would be responsible of issuing or tracking electronic currency. Instead, security in Nugglets is based on tamper protected hardware modules, which are assumed to be available and used in devices.

Trust management systems have also been proposed for incentive in ad hoc networks. In these systems, when a node detects an uncooperative node, it reports this observation to other peers. Peers may then decide not to cooperate with this uncooperative node. Challenges in trust management systems include, as noted e.g. in [18], vulnerability for false reports, complex decision algorithms, and additional signaling. In the context of P2P networks, evaluating nodes cooperative level, i.e. detecting bad behavior, is more challenging as provided contribution depends on factors which are not present in ad hoc networks such as distance between nodes or low capacity communication links. Hence, a peer may be determined to be an uncooperative one despite its willingness to contribute.

7. Conclusions and future work

The P2P credit system provides an incentive for peers to contribute. As the system is centralized and user-identity based it is suitable for users with both fixed and mobile devices. In this journal paper, we explored security challenges and mechanisms for the credit system. The paper extends our ICIW 2008 conference paper [1] with more extensive security analysis and literature survey. A survey and classification of threats against incentive mechanisms was provided. Then, we surveyed requirements and mechanisms for securing the credit system and presented an implementation of the credit system for mobile devices with BitTorrent P2P clients. However, the implementation is not tied to the BitTorrent protocol. In the future, the credit system could support other P2P clients with different protocols.

Every identified attack against the system cannot be prevented. A fundamental security problem in P2P networks is that information coming from individual peers cannot be trusted. This means that with some efforts, an attacker may gain illegitimately credits. However, a reasonable security level can be achieved with a combination of various security mechanisms. At the minimum, architecture must enforce that credits are used and collected in legitimate manner. In practice this requires that contribution is given only for peers with enough credits and that contribution is verified in authentic, preferably in privacy preserving, manner. Also, additional security level may be achieved with misbehavior detection mechanisms as well as with software tamper protection mechanisms.

The effect of the incentive mechanism is that it will make more contributions available. However, it is unclear will this additional contribution justify the overhead, which the related security processing and signaling causes. In the future, this question should be studied with user studies and field trials.

8. References

[1] Jani Suomalainen, Anssi Pehrsson, and Jukka K. Nurminen. A Security Analysis of a P2P Incentive Mechanism for Mobile Devices. The Third International Conference on Internet and Web Applications and Services (ICIW 2008), 2008.

[2] Olli Karonen and Jukka K. Nurminen. Cooperation Incentives and Enablers for Wireless Peers in Heterogeneous Networks. IEEE CoCoNet Workshop Cognitive and Cooperative Wireless Networks, 2008.

[3] John Douceur. The Sybil Attack. International Workshop on Peer-to-Peer Systems, 2002.

[4] Ravi Sandhu and Xinwen Zhnag. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. Symposium on Access Control Models and Technologies, 2005.

[5] Jari Arkko, Pekka Nikander, and Mats Näslund. Enhancing Privacy with Shared Pseudo Random Sequences. International Workshop on Security Protocols, 2005.

[6] Micheal Freedman and Rober Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. 9th ACM Conference on Computer and Communications Security, 2002.

[7] Ian Clarke and Oskar Sandberg. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Workshop on Design Issues in Anonymity and Unobservability, 2000.

[8] BitTorrent Protocol Specification. Version 1.0. September 2006. http://wiki.theory.org/BitTorrentSpecification. [Referenced April 4th 2009].

[9] Bram Cohen. Incentives Build Robustness in BitTorrent. Proceedings of the first Workshop on the Economics of Peer-to-Peer systems, 2003.

[10] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent? 4th USENIX Symposium on Networked Systems Design & Implementation, 2007.

[11] David Hales and Simon Patarin. How to cheat BitTorrent and why nobody does. University of Bologna Technical Report UBLCS-2005-12, May 2005.

[12] Hack KaZaA participation level – the easy answer. http://www.davesplanet.net/kazaa/. [Referenced April 4th 2009].

[13] Anirudh Ramachandran, Atish Das Sarma, and Nick Feamster. BitStore: An Incentive-Compatible Solution for Blocked Downloads in BitTorrent. The Economics of Networked Systems and Incentive-Based Computing in conjunction with ACM Conference on Electronic Commerce, 2007.

[14] Beverly Yang and Hector Garcia-Molina. PPay: Micropayments for Peer-to-Peer Systems. 10th ACM conference on Computer and communications security, 2003.

[15] David Hausheer and Burkahard Stiller. PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications. IFIP Networking Conference, 2005.

[16] Mira Belenkiy, Melisissa Chase, C. Chris Erway, John Jannotti, Alptekin Küpcü, Anna Lysyanskaya, and Eric Rachlin. Making P2P Accountable without Losing Privacy. ACM Workshop on Privacy In The Electronic Society, 2007.

[17] Levente Buttyan and Jean-Pierre Hubaux. Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. Swiss Federal Institute of Technology Technical report DCS/2001/001, 2001.

[18] Elgan Huang, Jon Crowcroft, and Ian Wassell. Rethinking Incentives for Mobile Ad Hoc Networks. The ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems, 2004.