Securing Identity Assignment Using Implicit Certificates in P2P Overlays

Juan Caubet, Oscar Esparza, Juanjo Alins, Jorge Mata-Díaz, Miguel Soriano

Securing Identity Assignment using Implicit Certificates in P2P Overlays

Juan Caubet, Oscar Esparza, Juanjo Alins,
Jorge Mata-Díaz, and Miguel Soriano

Universitat Politècnica de Catalunya (UPC)
SPAIN

Abstract. The security of the Peer-to-Peer (P2P) overlays networks has been questioned for years. Many works have been proposed to provide secure routing, anonymity, reputation systems, confidentiality, etc. However, the identity assignment has been less considered. These networks are designed so that each user has a unique identifier (nodeID), but the most of identity assignment systems allow malicious users to obtain a set of nodeIDs or even select certain identifiers. Thus, these users can disrupt the proper operation of a P2P overlay. In this paper, we propose a nodeID assignment protocol based on the issue of *implicit certificates*. Our purpose is to provide security services to struggle against the most of security threats in these networks with special attention to the identity assignment. This approach is based on the use of certificates and the joint generation of nodeIDs between a Certification Authority (CA) and the user. In addition, the use of implicit certificates presents certain advantages over the use of traditional certificates (*explicit certificates*).

Keywords: Identity Management; Implicit Certificates; Peer-to-Peer Overlays; Sybil Attack; Eclipse Attack

1 Introduction

Peer-to-Peer (P2P) overlay networks appeared a few years ago to solve routing problems of big distributed infrastructures, even for Internet-scale, as they can provide scalability, fault-tolerance, self-organization, and low-latency properties. According to the annual Cisco Visual Networking Index (VNI) Forecast [1], the P2P traffic represented around 30% of global IP traffic in 2011 and will grow at a compound annual growth rate (CAGR) of 23 percent from 2010 to 2015. However, overlays are not being widely used for commercial applications as they have important security threats.

P2P video streaming applications are the best example of commercial applications which need a special attention on security. Video streaming in P2P networks emerged as an evolution of sharing multimedia contents, initially for download. Their power to accommodate millions of users, together with their resilience to churn, reliability, and low cost are some of the reasons why these

networks are being increasingly used for these applications. Video-on-Demand (VoD) applications will produce three times more traffic by 2015 [1]. It is estimated that the amount of VoD traffic in 2015 will be equivalent to 3 billion DVDs per month. SopCast, PPTV, CoolStreaming, TVUnetworks and Zattoo are some of the many streaming applications and services that have been developed so far. However, most of them are proprietary video streaming and distribution platforms, which use the second generation of P2P networks[1] or distribute contents with poor access control and/or security.

Taking into account that some P2P overlay features make these systems vulnerable against certain classes of attacks, if we want to use overlays to implement commercial applications (like paid VoD services), it is necessary to solve a series of security problems. Moreover, this measure will also help users to be more confident about the reliability of these networks, who often tend to think that P2P networks are insecure by nature.

P2P overlays have been analyzed in depth to guarantee scalability and efficiency. However, few security mechanisms are being applied today. Most of these networks assume that nodes behave honestly, but this assumption is not acceptable in open environments. The existence of anonymous nodes and the lack of a centralized authority capable of monitoring (or punishing) nodes make these systems more vulnerable against selfish or malicious behaviors. These improper usages cannot be faced only with data confidentiality, nodes authentication, non-repudiation, etc. In particular, P2P overlays should follow the secure routing primitives described by Wallach in [2], which are: (1) secure maintenance of routing tables, (2) secure routing of messages, and (3) secure identity assignment to nodes. But the first two problems depend in some way on the third one. If the identity of the nodes within the overlay (nodeIDs) can be chosen by users without any control, we can have security and operational problems. Unfortunately, little attention has been paid so far to the way that nodeIDs should be constructed, or how to make access control mechanisms more robust. Like any other network or service, P2P overlay networks require a robust access control to prevent potential attackers join the network. Moreover, a robust identity assignment system is necessary to improve the users' confidence in P2P overlays, so they can use them for commercial applications. Nowadays, the most of these networks work pretty well without the need to assign nodeIDs in a secure way because they provide free services without Quality of Service (QoS) agreements. Thus users are willing to assume certain shortcomings and even an intrinsic risk in using such applications.

For these reasons we propose the use of digital certificates to provide security services such as authentication, confidentiality, integrity, etc., and a new protocol to assign identities leveraging the issuance of these certificates. We use *implicit certificates* [3, 4] since they presents certain advantages over the use of traditional certificates (*explicit certificates*). Implicit certificates are smaller and faster than explicit certificates. They do not include the issuer's signature within the certificate and require less computing time, there is less work involved extracting

[1] Structured P2P overlays belong to the third generation of P2P networks.

the public key of the sender than there is to verify a digital signature. Moreover, implicit certificates' generation enables us to easily construct secure nodeIDs. We propose to use nodeIDs jointly constructed by users and the Certification Authority (CA) to avoid Eclipse attacks [5], among other problems.

The rest of the paper is organized as follows: Section 2 summarizes some identity problems that arise when certain kinds of nodeIDs are used. Section 3 presents some proposals which attempt to prevent, detect and/or limit the identity problems experienced by these networks. Section 4 explains what the implicit certificates are and presents the notation used along the paper. Section 5 introduces our protocol to assign identities in a secure way in P2P overlays. Finally, some conclusions are drawn in Section 6.

2 Identity Problems in P2P Overlays

Most typical P2P overlay networks [6–9] are implemented using a Distributed Hash Table (DHT), which stores $\{key, value\}$ pairs together with the nodeIDs creating a virtual space. A *value* can be a certain resource (for instance a file), or the way to reach this resource in the overlay (a pointer), and the associated *key* is used to locate this resource into the network. The DHT is divided in subtables, which correspond to a certain zone of the virtual space, and are assigned to different nodes. So each node is responsible for one zone, and hence it is responsible for the {key,value} pairs contained in that zone (storing content and routing messages). Usually, a zone is assigned to a node whose nodeID is numerically close to the key values stored in the corresponding subtable of the DHT. Therefore, the location of the nodes in the virtual space is directly related to their nodeIDs. Unfortunately, in most current P2P overlay networks these identifiers are generated by the nodes locally. This means that users can choose their nodeIDs.

Users in a CAN network [6] are identified by their assigned zone within the virtual space, zones selected by them freely. In Chord [7] and Kademlia [8], nodeIDs are generated by the users using a hash function over their IP address. Pastry nodeIDs [9] are assigned randomly by the client software. And similarly in other overlay networks.

Several identity-related problems arise with the uncontrolled assignment of nodeIDs: Sybil attacks, Eclipse attacks, Man-In-The-Middle (MITM) attacks, the presence of whitewashers, etc.

2.1 The Sybil Attack

The management of multiple nodeIDs (Sybils) by the same (malicious) node simultaneously is known as Sybil attack [10]. Carrying out this attack, a malicious user can increase her presence within the overlay by artificially simulating the existence of several nodes. Thus, the attacker can manage a majority of colluding virtual nodes, which could damage the proper operation of the P2P network. For instance, an attacker performing the Sybil attack can improve its own reputation by using good feedback that comes from fake identities.

2.2 The Eclipse Attack

The Eclipse attack [5] is a way of routing poisoning which aims to separate a part of the P2P overlay network from the rest. The attacker tries to intercept all the messages directed to a specific node (or resource) by means of a set of nodes with nodeIDs numerically close to the nodeID of the target node (or the resource's value). Therefore, all messages will be routed across the attacker nodes and the correct nodes will be eclipsed from the view of each other.

2.3 The Man-In-The-Middle (MITM) Attack

As its name implies, in this attack, the attacker locates herself undetected between two target nodes with the purpose of spy on their communications or even manipulate them. Therefore, if nodeIDs (node placement in the virtual space) of the P2P overlay networks are assigned without any control, these networks will be extremely vulnerable to the MITM attacks.

2.4 Other Threats

The efficiency of the routing algorithms is based on the uniform distribution of the nodeIDs; therefore the overlay performance can be globally degraded whether the most nodeIDs belong to a particular area of the virtual space. If nodeIDs can be selected by the users, nobody will be able to ensure that these identifiers are uniformly distributed.

Other security threat related to the identity of the nodes is the presence of whitewashers within a P2P overlay network (nodes that purposefully leave and rejoin the network with a new nodeID in an attempt to shed any bad reputation they have accumulated under their previous nodeIDs [11]). Reputation systems can be used to prevent malicious behaviors and to promote honest collaboration among nodes. However, the effectiveness of these systems just depends on the stability of the nodeIDs. If a node can leave the network and rejoin it with a new (or different) nodeID, its accumulated reputation (good or bad) is removed.

3 Related Work

In this Section we discuss some research works that attempt to tackle some of the previous attacks or threats.

3.1 Centralized Solutions

Douceur, in [10], comments the impossibility to know if two nodes are managed by two real identities, or if there is only one user managing them, even asking other nodes within the overlay. Finally, he concludes that a trusted entity that certifies nodeIDs is the only solution to completely avoid the Sybil attack.

However, he is the first to suggest methods for imposing computational cost on creating identities and system conditions to mitigate the Sybil attack.

In [12], Castro et al. propose two centralized ways to generate nodeIDs. The first one is to delegate the identity assignment problem to a set of trusted CAs, which sign certificates that bind a random nodeID to a public key and an IP address. They assume that every node in the network has a static IP address and allow multiple certificates per IP address, which is an important drawback. And the second proposal is to charge money for certificates or bind nodeIDs to real-world identities in order to mitigate the Sybil attack. However, both solutions may not be liked by the users. The first option would not be suitable for free services, and with the second one, the users would lose their anonymity.

Srivatsa and Liu propose the use of certificates with a short life-time issued by a bootstrap server which also generates random nodeIDs [13]. This technique limits the number of nodeIDs that an adversary can obtain during a time period, depending on the life-time of the certificates, and maintains complete anonymity of the nodes. However, the life-time can affect the security of the system. If it is too short, the server can become a bottleneck as the update of the certificates introduces a significant computational overhead. On the other hand, longer life-times can cause greater exposure to compromise. Therefore, it is very important to set the system parameters taking into account this trade-off between security and cost.

In [14], Butler et al. consider the use of identity-based encryption (IBE) to improve the critical assignment of user identities in P2P overlay networks, where users' public keys are directly derived from their nodeIDs, which are calculated randomly by a Trusted Authority (TA). In this proposal a single host plays the role of both TA and bootstrap node and the node authentication is performed via callback using their IP addresses, main drawback of the scheme.

Baumgart and Mies propose to use a hash function over a public key to generate the nodeIDs [15]. Of course, the signature's public key must be additionally signed by a CA. Thus, this signature impedes the Sybil attack in the bootstrapping phase. In the absence of the TA, they propose to use a crypto puzzle to impede the Sybil and Eclipse attacks.

In [16, 5, 17], Aiello et al. propose to involve a human interaction with a centralized server in the authentication procedure. They use the OpenID protocol to authenticate users and a trusted entity which binds the user identity to the user public key and to a random 160 bit nodeID. Thus, the automatic nodeID generation by a user is impossible. However, the OpenID based authentication may not be a problem for an attacker with several OpenID accounts. In addition, the random generation can also create problems.

3.2 Distributed Solutions

In [12], Castro et al. propose bind nodeIDs to IP addresses in a distributed way and using cryptographic puzzles. Users must choose a key pair so that the hash of the public key has the first p bits to zero, which will be their nodeID. Then, to bind the nodeIDs with IP addresses authors define a computational

challenge that limits the number of nodeIDs obtained by the users. However, the computational cost normally is not a problem for attackers since they usually have enough computing power; not so the normal users, which may be using mobile devices such as smartphones, tablets, or netbooks. Moreover, IP addresses are vulnerable to IP Spoofing attacks or can even be dynamically assigned.

In the same line that Castro et al., a cryptographic puzzle mechanism has also been proposed by Rowaihy et al. to limit Sybil attacks [18]. Authors present an admission control system using a self-organized hierarchy of cooperative nodes and a chain of cryptographic puzzles. They exploit a hierarchical structure to distribute load and increase resilience to targeted attacks. They also propose to refresh the challenges constantly in order to avoid pre-computation. The obtained nodeID is a hash function over the node public key, previously selected by the user, and a random number generated by the root node. As with the above solution, this mechanism also negatively affects to the nodes that have limited resources, and it does not solve the problem because malicious hosts with enough resources can manage a large number of nodeIDs. The effectiveness of this solution depends on the cost and the degree of hardness of solving the puzzles. Moreover, if an attacker is a member of the hierarchy, she can take advantage of her position, as she will need a smaller number of puzzles to obtain a nodeID.

In [19], Da Costa et al. try to solve computing problems which affect honest nodes when they have to solve cryptographic puzzles. They propose the use of adaptive computational puzzles to limit the spread of Sybils. The proposal parameterizes the complexity of puzzles according to the nodes behavior. Users of the nodes whose behavior is more similar to the average behavior of the rest of the network are benefited with less complex puzzles. Otherwise, users are forced to solve more complex puzzles to obtain nodeIDs.

Lu proposes, in [20], a conundrum verification scheme which allows access to the P2P network through a more expensive process of identity acquisition. It works over a structured network with hierarchy; super nodes manage regions which include a lot of guard nodes and normal nodes. The solution is composed of two phases, the first where nodes join the network paying certain price (e.g., solve a cryptographic puzzle) and the second where the super nodes use the guard nodes to obtain the nodeIDs of the normal nodes and to verify the validity of these nodeIDs. This verification is performed based on the statistics result. The weakness of this solution is in the binding of the nodeIDs, selected by the users, with their IP addresses to verify the identities.

In [14], Butler et al. also develop two decentralized identity assignment protocols. A fully decentralized ID-based assignment scheme and an approach that retains the separation of duties in a decentralized model at a low cost by using a hybrid of ID-based and symmetric key cryptography. But in the same way as in the centralized protocol, nodes are weakly authenticated via callback using their IP addresses, which is insufficient to prevent the Sybil attack.

3.3 Social Network-Based Solutions

In [21], Yu et al. present *SybilGuard*, a social networks based protocol which limits the Sybil attacks. NodeIDs are represented as nodes in a graph and an edge between two nodeIDs indicates a human-established trust relationship. SybilGuard exploits the fact that all the malicious nodes created by a given physical attacker are only sparsely connected to the real social network. The edges connecting the honest and the Sybil regions are called attack edges, and its number is independent of the number of Sybil nodes. Each node constructs its own partition of the network using a procedure based on random routes for each of its edges, a special kind of random walk. When a node wants to verify that another node is honest, it checks for intersections in their routes. A node accepts a suspect node only if at least half of its routes intersect with any of suspect's routes, meaning that most likely that node will belong to the same (honest) region as it. SybilGuard allows accepting $O(\sqrt{n}\log n)$ Sybil nodes assuming up to $O(\sqrt{n}/\log n)$ attack edges, where n is the number of honest nodes.

Yu et al. propose, in [22], an update to SybilGuard, called *SybilLimit*. This proposal allows accepting $O(\log n)$ Sybil nodes assuming up to $O(n/\log n)$ attack edges.

Unfortunately, in terms of functionality, these solutions are not suitable for P2P overlay networks as there are several difficulties that complicate their real-world deployment. First, they require previous trustworthy relationships among nodes. Second, they require symmetric key sharing before the creation of links between nodes, as each social connection has a unique symmetric key. And finally, they force all nodes to store a different symmetric key with each of their friends.

In [23], Tran et al. propose Gatekeeper, an optimal distributed Sybil-resilient admission control protocol that significantly improves over SybilLimit. For the case of $O(1)$ attack edges, it admits only $O(1)$ Sybil identities in a random expander social networks. In the face of $O(n/\log n)$ attack edges, the protocol admits $O(\log n/\log n)$ Sybils per attack edge.

Lesueur et al. present, in [24], a sybilproof distributed identity management system based on invitations. Thus they rely on social relationships to prevent the Sybil attacks. Their scheme is based on a balanced tree that represents the social relationships between users. However they limit the number of invitations each member and each of the members he has invited comparing the members between themselves.

4 Background

4.1 Implicit Certificates

Implicit certificates [3, 4] are not standard public key certificates, such as X.509. A standard certificate explicitly contains the public key of the user and the signature of the CA that issued the certificate, together with more information. An implicit certificate does not contain either the owner's public key or the signature of the CA. It contains the needed information so that any interested

entity can calculate the associated public key; a reconstruction public parameter, the identity of the issuing CA, the owner's identity and the validity period of the certificate, at minimum. Then, an implicit certificate is simply a pair (Z, I), where Z denotes the reconstruction public parameter and I denotes the included information in the certificate. Note that Z is only an elliptic curve point, so implicit certificates have a shorter length than standard certificates.

The receiver of an implicit certificate constructs the associated public key using Z and the CA's public key before validating the sender's signature. In the same way that with explicit certificates, the receiver must trust the CA and the authenticity of the CA's public key in order to arrive at the assurance that the constructed public key is indeed sender's public key. With explicit certificates, the receiver verifies the signature of the sender with the CA's public key, and so she is sure that the public key belongs to the sender. However, solely the certificate is not enough to authenticate a user, as this certificate is public information. To authenticate herself, a user must demonstrate knowledge of her private key using a secure cryptographic protocol; a digital signature scheme, for instance. The same applies to implicit certificates, a sender must demonstrate knowledge of her private key. In that moment, the sender has authenticated herself and has also demonstrated that the used public key belongs to her. With implicit certificates, the authentication that the used public key belongs to sender and the authentication that the user is who she claim to be are not separable.

There are two types of implicit certificates [25]: (1) identity-based public key implicit certificates, where the CA generates the user's private key, and (2) self-certified public key implicit certificates, where the user selects her private key and the CA only knows the associated public key. In this paper, we only consider the self-certified public key implicit certificates.

4.2 Notation

The elliptic curve domain parameters used in this paper are denoted as follow. Let q denotes the order of the underlying finite field F_q, let E be an elliptic curve defined over F_q. Let G denotes a base point in $E(F_q)$, and let n denotes the order of G, assuming that n is prime. Thus $nG = \mathcal{O}$ and $G \neq \mathcal{O}$. We also assume that the discrete logarithm problem in the group $\langle G \rangle$ of points generated by G is intractable. More accurately, there is no probabilistic polynomial-time algorithm (polynomial in the security parameter $l = \lfloor log_2\ n \rfloor$) which on input $C \in_R \langle G \rangle$, $C \neq \mathcal{O}$, can output $c \in [1, n-1]$ satisfying $C = cG$ with non-negligible probability. Let $c \in [1, n-1]$ be the private key of the CA, and let $C = cG$ be its public key. Let a user be denoted by i. The CA generates a signature for the U_i's implicit certificate (s'_i). This certificate contains at minimum the identity of the user (U_i), the identity of the CA and the validity period of the certificate; I_i. Table 1 presents a global summary of the used notation along the paper.

Table 1. Notation

U_i	The identity of the user i.
P_i	The pseudonym of U_i within the overlay (nodeID).
I_i	Some information that is included in U_i's certificate.
PV_i	The private key of U_i.
PB_i	The public key of U_i.
c	The private key of the CA.
C	The public key of the CA.
s'_i	The signature of the CA for U_i.
Z_i	The U_i's reconstruction public data.
$r_i,\ v_i,\ w_i$	Private parameters of the U_i's request.
$R_i,\ s_i,\ K_i,\ T_i,\ W_i,\ F_i$	Public parameters of the U_i's request.
$H(x)$	A secure hash function on x.
M	An information message.
m	A complete message which contains M.
$Sig_y(x)$	A signature on the message x using the key y.

5 Identity Assignment Protocol

First of all, it is important to state that there are many P2P-based services that are widely known and used by Internet users, some of them based on P2P overlay networks. These P2P overlay services have been working pretty well until now without the need to assign nodeIDs in a secure way. This is probably due to the fact that the most of these services are free, there are no Quality of Service (QoS) agreements, and the same users assume certain shortcomings and even an intrinsic risk in using such applications. However, if we want to use such networks to provide commercial applications, it is mandatory to solve some security vulnerabilities, starting with the identity assignment problem, as stated by Wallach in [2]. For this reason, we have designed a new protocol, based on the issuance of self-certified implicit certificates, to assign nodeIDs in a secure way.

Our protocol allows users to generate nodeIDs but under the supervision and participation of a CA. NodeIDs are generated using the user's public key as input of a hash function, such as SHA-1. The CA controls that users generate their certificates correctly, but without knowing the associated private key of them. However, users cannot decide unilaterally what will be that private key; the CA takes part in the generation process but without knowing the final value. In this way, nodeIDs are very easy to verify but cannot be generated by a user unilaterally. Note that the user's nodeID are not stored in the certificate, it must always be calculated from the user's public key.

5.1 Design Requirements

We consider that a robust identity management system must meet the following requirements:

- *Uniqueness:* each user (real identity) should only manage one node (nodeID) in the system. This requirement is necessary to limit the Sybil attacks. Using a Trusted Third Party (TTP) is the best way to limit this attack.
- *Stability:* nodes should not have the possibility to change their nodeIDs uncontrollably. This requirement is necessary to implement efficient reputation systems within the overlay.
- *Joint Management:* neither the nodes themselves nor a TTP should be able to choose the users' nodeIDs unilaterally. This requirement is necessary to avoid the Eclipse attacks.
- *Uniformity:* nodeIDs should be uniformly distributed in the virtual space. This requirement is necessary to achieve the proper load-balancing among all nodes of the overlay.
- *Verifiability:* all nodes should be able to check if certificates and nodeIDs have been properly generated and bound.
- *Revocability:* any user should be able to revoke her certificate in case of losing or seeing compromised her private key, and to get a new one maintaining its nodeID. In addition, if a dishonest or malicious node's behavior within the overlay is detected, the CA should be able to revoke her certificate (and nodeID) and thus prevent the user can access to the overlay again.
- *Traceability:* if a user commits an illegal action within the overlay and she must be judged for it, authorities should have the possibility to trace this user and match the nodeID with the user's real-world identity.

However in this paper we only propose an identity assignment protocol which guarantees the above requirements long as the CA properly controls the user access.

5.2 Protocol Specification

Next we describe our protocol in detail. Figure 1 shows the data exchanged between the two parties (CA and user) and the operations carried out by them. Note that we do not define what encryption and digital signature algorithms should be used; as many options can be chosen.

Step 1: First of all, the CA and the user establish a secure communication to exchange sensitive information. Then, it generates a private parameter $r_i \in [1, n-1]$ and a public parameter $R_i = r_i G$ for this request. And finally, the CA sends that public parameter to U_i together with the information that she must include in the certificate (I_i).

Step 2: U_i receives the public parameter generated by the CA and generates two private and three public parameters. $\{v_i, w_i\} \in [1, n-1]$ are the private parameters and $W_i = w_i G$, $K_i = v_i R_i$, $T_i = v_i w_i C$ and $F_i = v_i^{-1} G$ are the public parameters. Then, she calculates the reconstruction public parameter as $Z_i = K_i + T_i$ and the hash value of that parameter concatenated with the certificate information (I_i) as $h_i = H(I_i || Z_i)$. Finally, U_i calculates $s_i = h_i w_i + v_i^{-1} \ mod \ n$ and provides the CA with $\{s_i, K_i, T_i, W_i, F_i\}$.

Step 3: The CA receives the data sent by U_i, calculates Z_i and h_i, and verifies that $s_iG = h_iW_i + F_i$. Thus, it ensures that the received parameters are valid. Then, the CA generates the signature $s'_i = h_ir_i + s_ic \ mod \ n$ and sends it to U_i.

Step 4: U_i receives the signature and verifies that $s'_i = h_iR_i + s_iC$ and generates her private key $PV_i = v_is'_i \ mod \ n$. Finally, U_i generates her public key $PB_i = PV_iG$ and calculates her nodeID as the hash value of PB_i ($P_i = H(PB_i)$). Now, the public parameter s_i, the private parameters v_i and w_i, and the signature s'_i can be deleted.

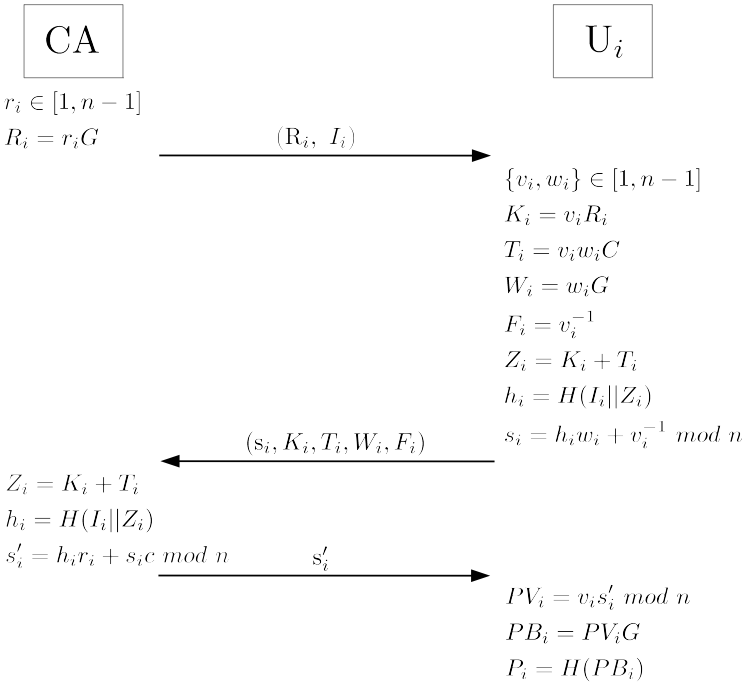

Fig. 1. The nodeID generation scheme.

5.3 Public Key Generation and nodeID Validation

Step 1: Whenever a user (U_i) is going to send a message (M), she includes her certificate's information (I_i) together with the reconstruction public parameter (Z_i). By now U_i has the following message: $m = M||I_i||Z_i$. Finally, she signs m using her private key (PV_i) and sends $m||Sig_{PV_i}(m)$ to the receiver. Note that for simplicity we have only signed the message, not encrypted.

Step 2: Now, the receiver can authenticate the sender and validate her nodeID and her signature as follows:

1. Calculates $h_i = H(I_i||Z_i)$.
2. Generates the sender's public key as $PB_i = h_i Z_i + C$.
3. Validates the sender's nodeID by computing $P_i = H(PB_i)$.
4. Validates the message's signature using PB_i.

Note that $PB_i = PV_i G = h_i v_i R_i + h_i w_i v_i C + C = h_i K_i + h_i T_i + C = h_i Z_i + C$.

5.4 Performance Analysis

We have attempted to develop a secure identity assignment protocol to avoid the most threats related to the identities in P2P overlays, but bearing in mind that all security measurements have a cost. Regardless of the system complexity, the overhead, delay, computational cost, and so on, introduced by the protocol only affects the first time that a user joins the overlay (or each time that a user must update her overlay certificate). For this reason we have defined a generic protocol which can be implemented using any encryption and digital signature algorithms.

Regarding the use of implicit certificates, it is important to note that this decision involves an improved performance. Implicit certificates also contain identification data, but the user's public key and issuer's signature can be considered to be combined into a single element, the reconstruction public parameter. This substantially reduces the number of bytes that a sender needs to send a receiver, since the CA's signature no longer needs to be sent (64 bytes if an elliptic curve P-256 is used).

Regarding the processing overhead, processing a message with a standard certificate requires two signature validations: one for the signature on the certificate and the other for the signature on the received message. Using implicit certificates there is only one signature validation for the signature on the received message. However, first of all, the receiver needs to calculate the public key using one elliptic curve scalar multiplication operation and one point addition operation. In the Elliptic Curve Digital Signature Algorithm (ECDSA), for instance, are needed three modular operations, two elliptic curve scalar multiplication operations and one point addition operation. Therefore, the three modular operations are not necessary if we use implicit certificates.

Moreover, this protocol ensures that all nodeIDs are uniformly distributed in the virtual space of the network, which ensures the proper load-balancing among all nodes of the overlay. We achieve this requirement by using a hash function to generate identifiers.

5.5 Security Analysis

In Public Key Infrastructures (PKIs) based on implicit certificates, a user must demonstrate the knowledge of a private key (PV_i) by using a signature algorithm, such as ECDSA. Since generating the associated public key (PB_i) is not

a sufficient proof that the public key is authentic. In our protocol, PV_i is generated using the CA's signature s'_i. And this signature are calculated using the CA's private key c and the private parameter r_i. PB_i is generated using the CA's public key CA and the reconstruction public parameter that is provided by the user. Thus, if a user validates a signature using PB_i, she can be sure that the correct CA's public key has been used in constructing PB_i. Otherwise, the signature validation will fail. The same occurs when a different reconstruction public parameter is used.

However, if an attacker wants to find the private parameters v_i and w_i, she must solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) in Z_i. Unfortunately for the attacker, solving the ECDLP is computationally hard. We have also said before that the user can to delete certain parameters to finish the process, concretely s_i, s'_i, v_i and w_i. Thus we will prevent an attacker can find the CA's private key c if she sneaks into a user's computer. For example, if an attacker knows $\{v_i, w_i\}$, then she can calculate s_i. Furthermore, if the attacker knows s_i and s'_i for at least two users, then she can find c and r_i.

Regarding the identity assignment, using this protocol we can guarantee two security related requirements; verifiability and joint management (Section 5.1). Any node can check if a certificate or nodeID has been properly generated and bound, and no user can generate a valid certificate or nodeID without the supervision and participation of a CA.

6 Conclusions

Vulnerability to certain attacks is a strong obstacle to develop certain services (e.g. commercial applications) in P2P overlay networks. In this paper we have proposed a secure identity assignment scheme with the aim of solving some of these vulnerabilities and turning these networks into a powerful platform for commercial applications. Our proposal issues identities in a secure and anonymous way without affecting the current operation of the overlays. It neither allows the users to select their nodeIDs nor the CA to select the users' nodeIDs, ensuring that users are placed in virtual space pseudo-randomly and uniformly distributed. In addition, the anonymity of users within the overlay may remain secured. Finally, we have analyzed the performance and the security of our scheme, and obviously there is a trade-off between both aspects, security and performance. However, if we consider that a user only executes the protocol the first time she wants to join the network, the user Quality of Experience (QoE) will not be affected by the use of our protocol. And regarding security, our proposal has only one weakness; we must trust the CA. But using a CA is the only way to avoid 100% certain attacks (Sybil attack, Eclipse attack, etc.).

Future work will be focused on proposing an identity management system to provide traceability of malicious users and revocation of their certificates and nodeIDs.

Acknowledgements

This work was supported partially by the Spanish Research Council with Project SERVET TEC2011-26452, by the Spanish Ministry of Science and Education with Project CONSOLIDER CSD2007-00004 (ARES) and by Generalitat de Catalunya with Grant 2009 SGR-1362 to consolidated research groups.

References

1. Cisco visual networking index: Forecast and methodology, 2011-2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.
2. Dan S. Wallach. A survey of peer-to-peer security issues. In *International conference on software security: theories and systems (ISSS)*, volume 2609 of *LNCS*, pages 42–57, Tokyo, Japan, november 2002. Springer-Verlag Berlin, Heidelberg.
3. Leon A. Pintsov and Scott A. Vanstone. Postal revenue collection in the digital age. In *4th International Conference on Financial Cryptography*, volume 1962 of *LNCS*, pages 105–120, Anguilla, BWI, february 2000. Springer Verlag.
4. Daniel R. L. Brown, Robert Gallant, and Scott A. Vanstone. Provably secure implicit certificate schemes. In *Proceedings of Financial Cryptography*, volume 2339 of *LNCS*, pages 156–165. Springer Berlin Heidelberg, 2002.
5. Romano Fantacci, Leonardo Maccari, Matteo Rosi, Luigi Chisci, Luca Maria Aiello, and Marco Milanesio. Avoiding eclipse attacks on kad/kademlia: an identity based approach. In *IEEE International Conference on Communications (ICC)*, pages 1–5, Dresden, Germany, june 2009. IEEE Press.
6. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, pages 161–172, San Diego, CA, USA, august 2001. ACM New York, NY, USA.
7. Ion Stoica, Robert Morris, David Karger, M. Frans Kaaskoek, and Hari Balakrishman. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, pages 149–160, San Diego, CA, USA, august 2001. ACM New York, NY, USA.
8. Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *1st International Workshop on Peer-to Peer Systems (IPTPS)*, pages 53–65, Cambridge, MA, USA, may 2002. Springer-Verlag London, UK.
9. Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms*, volume 2218 of *LNCS*, pages 329–350, Heidelberg, Germany, november 2001. Springer-Verlag London, UK.
10. John R. Douceur. The sybil attack. In *1st International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 251–260, Cambridge, MA, USA, may 2002. Springer-Verlag London, UK.
11. Sergio Marti and Hector Garcia-Molina. Taxonomy of trust: Categorizing p2p reputation systems. *Comput. Netw.*, 50(4):472–84, 2006.

12. Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. *ACM Operating Systems Review (OSR)*, 36:299–314, 2002.
13. Mudhakar Srivatsa and Ling Liu. Vulnerabilities and security threats in structured overlay networks: a quantitative analysis. In *20th Annual Computer Security Applications Conference (ACSAC)*, pages 252–261, Tucson, AZ, USA, december 2004. IEEE Computer Society Washington, DC, USA.
14. Kevin R.B. Butler, Sunam Ryu, Patrick Traynor, and Patrick D. McDaniel. Leveraging identity-based cryptography for node id assignment in structured p2p systems. *IEEE Transactions on Parallel and Distributed Systems*, 20(12):1803–1815, 2009.
15. Ingmar Baumgart and Sebastian Mies. S/kademlia: A practicable approach towards secure key-based routing. In *13th International Conference on Parallel and Distributed Systems*, volume 2, pages 1–8, Washington, DC, USA, december 2007. IEEE Computer Society.
16. Luca Maria Aiello, Marco Milanesio, Giancarlo Ruffo, and Rossano Schifanella. Tempering kademlia with a robust identity based system. In *8th International Conference on Peer-to-Peer Computing (P2P)*, pages 30–39, Washington, DC, USA, september 2008. IEEE Computer Society.
17. Luca Maria Aiello, Marco Milanesio, Giancarlo Ruffo, and Rossano Schifanella. An identity-based approach to secure p2p applications with likir. *Peer-to-Peer Networking and Applications*, 4:420–438, 2011.
18. Hosam Rowaihy, William Enck, Patrick D. McDaniel, and Thomas F. La Porta. Limiting sybil attacks in structured p2p networks. In *26th IEEE International conference on Computer communications (INFOCOM)*, pages 2596–2600, Anchorage, Alaska, USA, may 2007. IEEE Communications Society.
19. Weverton Luis Da Costa Cordeiro, Flávio Roberto Santos, Gustavo Huff Mauch, Marinho Pilla Barcelos, and Luciano Paschoal Gaspary. Identity management based on adaptive puzzles to protect p2p systems from sybil attacks. *Comput. Netw.*, 56(11):2569–2589, july 2012.
20. Chuiwei Lu. Detection and defense of identity attacks in p2p network. In *4th International Symposium on Advances in Computation and Intelligence (ISICA)*, volume 5821 of *LNCS*, pages 500–507, Huangshi, China, october 2009. Springer-Verlag Berlin, Heidelberg.
21. Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham D. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking*, 16(3):576–589, 2008.
22. Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. *IEEE/ACM Transactions on Networking*, 18(3):885–898, 2010.
23. Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman S.M. Chow. Optimal sybil-resilient node admission control. In *30th IEEE International Conference on Computer Communications (INFOCOM)*, pages 3218–3226, Shanghai, P.R. China, april 2011. IEEE Communications Society.
24. François Lesueur, Ludovic Mé, and Valérie Viet Triem Tong. A sybilproof distributed identity management for p2p networks. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 246–253, Marrakech, Morocco, july 2008. IEEE Computer Society.
25. Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.