Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems

Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel
Systems and Internet Infrastructure Security Laboratory
Department of Computer Science and Engineering
Pennsylvania State University, University Park, PA 16802
{ryu, butler, traynor, mcdaniel}@cse.psu.edu

Abstract

Structured peer-to-peer systems have grown enormously because of their scalability, efficiency and reliability. These systems assign a unique identifier to each user and object. However, current assignment schemes allow an adversary to carefully select user IDs and/or simultaneously obtain many pseudo-identities—leading ultimately to an ability to disrupt the P2P system in very targeted (and dangerous) ways. In this paper, we propose novel ID assignment protocols based on identity-based cryptography. This approach permits the acquisition of node IDs to be tightly regulated without many of the complexities and costs associated with traditional certificate solutions. We broadly consider the security requirements of ID assignment and present three protocols representing distinct threat and trust models. A detailed empirical study of the protocols is given. Our analysis shows that the cost of our identity-based protocols is nominal, and that the associated identity services can scale to millions of users using a limited number of servers.

1. Introduction

Peer-to-peer networks are now ubiquitous. They provide a resilient media for the efficient storage and retrieval of file objects. Such models change the nature of storage and provide a vector toward dynamic and massively distributed global information sharing. However, while the object sharing techniques have advanced rapidly, security services protecting this media have yet to mature. This is largely due to a highly diverse, untrusted, and often anonymous user community.

Structured P2P systems assign a unique key identifier (ID) to every object and node. IDs associated with objects are mapped by P2P overlay protocols to the node responsible for that object. The assignment of node IDs is therefore critically important to the efficiency and security of the peer-to-peer system. However, current peer-to-peer systems use node ID assignment techniques that can be trivially manipulated by an adversary. Proposed solutions to these problems largely rely on the use of trusted certificate authorities and a structured public-key infrastructure (PKI) to assign and certify node IDs. These schemes, however, require maintenance of complex PKI systems, which can be difficult or infeasible to implement in practice.

In this paper we consider the use of identity-based cryptography to assist in the security and performance critical assignment of user identities in peer-to-peer systems. This approach avoids many of the complexities of PKI usage (a user's public key is directly derivable from their identity), and reduces the overheads associated with authentication. We exploit these features in peer-to-peer systems by assigning an ID and providing the associated identity-based private key to each joining node.

We identify three protocols representing diverse trust models and performance profiles based on identity-based cryptography: a decentralized scheme (protocol 1), a centralized scheme (protocol 2), and a hybrid of two approaches (protocol 3). We have built functional ID client and server implementations and tested them in our laboratory environment.

Our empirical analysis considers the relative performance of the protocols and their scalability. We found that we could achieve protocol run-times similar to centralized solutions with a decentralized architecture using a hybrid symmetric and ID-based cryptographic approach. We also found that our protocols could scale easily, where 2 servers could conservatively sustain a community of over 600,000 nodes, and 50 ID servers could support over 15,000,000 nodes.

The rest of this paper is organized as follows. Section 2 gives a brief overview of structured P2P networks and identity-based cryptography, and identifies the broad goals

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

and assumptions of this work. Section 3 describes our novel protocols in detail. Section 4 describes an empirical evaluation of the proposed approach. Open problems and operational issues are discussed in Section 5. Section 6 discusses important related work and Section 7 concludes.

2. Background

This section presents relevant background in peer-to-peer systems and identity-based encryption, and describes the security and performance goals of our approach.

2.1. Structured P2P overlay protocols

Structured overlays are designed to allow for scalable, efficient and reliable object placement within a dynamic virtual topology. To generalize, every node and object in a peer-to-peer system is assigned a unique identifier (ID).[1] A node locates an object by mapping the *object key* (the object's ID) to a node ID responsible for that object. The responsible node then supplies the object directly or indicates where/how it can be acquired.

In the representative systems Chord [18], Pastry [14] and Tapestry [19], node IDs are deterministically assigned by hashing the host's IP address. Conversely, in CAN [12], every node randomly picks its own node ID upon entering the system. In these systems an adversary can carefully select identities (either directly or by IP spoofing) such that they become the responsible node for sensitive objects. In a related technique, an adversary mounts a *Sybil* attack by obtaining a large number of simultaneous identities [9]. These identities probabilistically interpose the adversary in the routing paths for a great many objects, and thus permitting it to disrupt or manipulate the search process [4, 16].

2.2. Identity-based cryptography

Public keys in identity-based public key cryptosystems are simple data objects [2, 15], e.g., ASCII string email address. A trusted third party, called the *private key generator* (PKG), generates the corresponding private key using secret information associated with the public parameters. Using this construct, anyone can encrypt messages or verify signatures without prior key distribution beyond the the dissemination of public parameters and the public key "strings". This is useful where the deployment of a traditional certificate authority-based public key infrastructure is inconvenient or infeasible.

A central operational consideration of ID-based cryptography is that the private keys must be obtained from the PKG. How one securely and efficiently obtains this private key is essential to the security of the supported system. For example, how the PKG decides who should be given the private key associated with an email address is crucial to maintaining the integrity of the system. Another consideration is cost: key generation can be computationally expensive (see Evaluation in Section 4).

[1] These identities are transient pseudonyms for the real users, and hence are often referred to as "pseudo-identities". We use the terms identity and pseudo-identity interchangeably throughout.

2.3. Protocol Setup

This work is focused on the secure assignment and authentication of pseudo-identities in peer-to-peer systems. As such, we define the following goals of the system:

- *Secure ID assignment* - each user must be given a unique pseudo-identity (or just "identity" throughout) to which he can later be authenticated. The user must not be able to influence the content of that ID in any way, e.g., she cannot select or predict the ID.
- *Sybil attack mitigation* - the number of simultaneous pseudo-identities a node can acquire should be bounded by the system.
- *Pseudo-identity authentication* - other participants should be able to authenticate all users (nodes) in the system.
- *Limited overheads* - the costs associated with use of the IDs should be nominal.
- *Simplicity* - the complexity of the creation, maintenance, and use of the system should be low.

Discussed more fully in the protocols that follow, each joining node is weakly authenticated via callback: all responses to requests are transmitted through a server-initiated TCP connection (see protocols for details). Further, each such communication is protected by a secure channel established via a Diffie-Hellman exchange [8]. We further assume the existence of some loosely synchronized secure clock.

The following notation is used throughout:

IP_A : node A's IP address
ID_A : node A's ID assigned in the overlay network
K_A^+, K_A^- : node A's public key and private key
$K_{A \cdot B}$: shared secret key between node A and node B
$E(m,k)$: encryption of message m using the key k
$HMAC(m,k)$: keyed-hash message authentication code of message m using the key k
$Sign(m,k)$: signature of message m using the key k
TS_i : time stamp
$a \parallel b$: concatenation of two strings, a and b

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

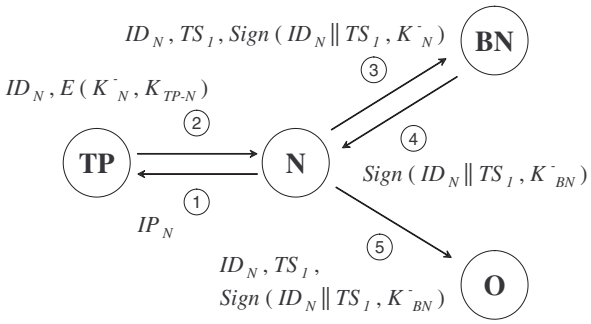

Figure 1. Node join in protocol 1

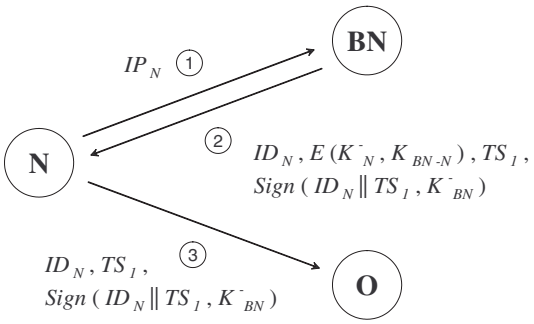

Figure 2. Node join in protocol 2

3. Protocol Specification

In the following subsections, we present three protocols that authenticate node IDs and protect structured peer-to-peer networks against Sybil attacks. We describe each protocol's specification and operation, and briefly discuss the tradeoffs inherent to each approach.

3.1. Protocol 1 : Trusted Third Party

In the first protocol, the binding between a node's ID and its private key is performed by a *trusted third party*, as shown in Figure 1. Here, the trusted third party assigns random node IDs and generates the corresponding private keys through ID-based cryptographic techniques.

In order for a node N to join an overlay network, it first contacts the trusted third party TP[2] and provides its IP address. After weakly authenticating its identity via callback, TP gives a randomly-generated ID and the corresponding private key. N then contacts the bootstrap node BN[3] and provides its ID and a timestamp, both signed with N's private key. Upon verifying a join request from N, BN returns a signed copy of the timestamp and node ID.

The Sybil attack is prevented because of the callback behavior: only if the node can be reached at the IP address given will it receive a response from the bootstrap node. Further, because BN has signed the response, it can be used as a token of authenticity. O can verify the signature, as BN's identity is known and hence, its associated public key is also known, due to the use of identity-based cryptography.

This protocol can noticeably reduce cost and system complexity compared to a traditional public key infrastructure, as it requires neither prior key distribution nor certificates. The decentralized nature of this architecture also provides for the separation of duties for policy and enforcement in the system.

[2]As with traditional centralized authorities, the procedure of requesting and transmitting private keys can be offline to reduce the possibility of revealing private keys generated by the third party.

[3]Finding the bootstrap node is application-specific. We assume that a new node joining the network knows initially about the bootstrap node that is already part of the system.

3.2. Protocol 2 : Trusted Bootstrap Node

In contrast with the previous scheme, the Trusted Bootstrap Node protocol shown in Figure 2 implements a centralized system. Specifically, the bootstrap node becomes the arbiter of network membership and trusted information.

When N attempts to join the network, it sends its IP address to BN. BN weakly authenticates N's identity through callback. Should N successfully demonstrate control over its claimed IP address, BN generates and assigns a node ID, a corresponding private key and a token to be used for authentication with member nodes in the network.

The major advantage of this protocol is the reduction in overhead associated with the interaction of a third party. This can simplify the procedure of joining a node, as the bootstrap node deals with both assigning node IDs and generating private keys.

3.3. Protocol 3 : Trusted Assignor Node

The previous two protocols trade off the separation of duties inherent to a decentralized architecture with the overall performance of a centralized scheme. The Trusted Assignor Node protocol is a hybrid of these two approaches. Specifically, a single bootstrap node generates only the private keys and delegates the authority of assigning node IDs to one of many trusted nodes.

Prior to operation, the bootstrap node selects the trusted nodes for assigning node IDs and establishes secret keys with them. When N attempts to join the network, as shown in Figure 3, it transmits its IP address to a trusted assignor node AS. After verifying the identity, AS generates the node ID and issues a timestamped token as proof of authentication. Upon verification of a token sent from N, BN

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

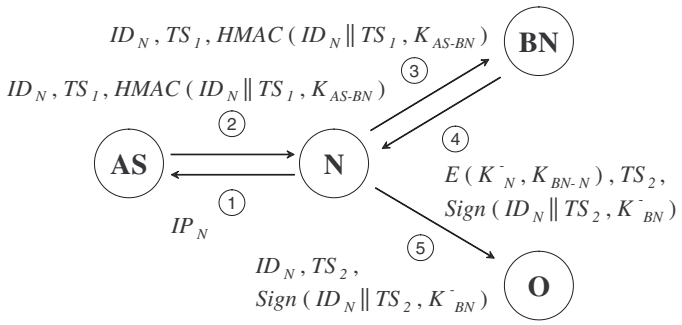

Figure 3. Node join in protocol 3

Operation	Cost	σ
Key creation	37.340	2.302
Node signature	80.722	3.548
Request verification	74.649	5.157
ID token creation	20.095	1.036
Symmetric-key token creation	0.131	0.052

Table 1. Cryptographic microbenchmarks

provides both a private key and a second token to be used for proving N's authenticity to O.

4. Evaluation

In this section we consider the cost of the three protocols. We have built an initial implementation written of all three protocols in C. All identity-based cryptographic algorithms use the pairing-based cryptography (PBC) library [10]. We parameterized the library to use supersingular elliptic curves over a non-random oracle construction [11] and Cha-Cheon signatures [5]. All experiments were executed using a dual processor G5 (server) and a Mac mini 1.5Ghz G4 (client), both of which were running the Apple OS X 10.4.7 operating system. All results reported below represent the average of 1,000 executions of the protocol or other measured function (in milliseconds).

4.1. Cryptographic Microbenchmarks

There are five significant cryptographic operations used in the protocols: the creation of the identity-based key (all protocols), the signing of the ID request (protocol 1), the verification of the node request (protocol 1), the creation of the ID-token (all protocols), and the creation of a symmetric key-based token (protocol 3). The measured costs are detailed in Table 1. The signature and subsequent verification operations are appreciably more expensive than the other operations. Such results are not unexpected, as they

	TP Exchange		BN Exchange	
Message	One	Two	Three	Four
Step	3.376	46.510	166.243	63.781
Cumulative	3.376	49.886	216.129	279.910

Table 2. Protocol 1 performance (in msecs)

	BN Exchange	
Message	One	Two
Step	1.750	115.319
Cumulative	1.750	117.069

Table 3. Protocol 2 performance (in msecs)

represent the most computationally intensive operations in identity-based cryptography.

4.2. Protocol Benchmarks

We now break down the per-flow and total costs for each of the protocols. Table 2 presents the results for the four messages composing protocol 1. In this, the third message (first message of the BN exchange) consumes about 60% of the total delay per protocol iteration - a result of both the client signature and subsequent signature verification.

Protocol 2 leads to a simplified performance analysis shown in Table 3. Note that the average execution time is less than half that of protocol 1. This is due to the fact that the single exchange eliminates a signature creation and verification. However, this efficiency has a cost: all server functions (and hence all trust) must be placed in a single authority. This may not be appropriate (or even feasible) in many environments.

Protocol 3 retains the separation of duties between the different servers while retaining low cost. For example, the AS exchange fulfills the same purpose as the the TP exchange in protocol 1 at $1/20$th the cost. This is achieved by applying symmetric key cryptography to secure communication between the ID authority and the bootstrap node. A cost analysis of this protocol is presented in Table 4.

There are also more efficient parameters for an ID-based cryptosystem. MNT elliptic curves, for example, are more

	AS Exchange		BN Exchange	
Message	One	Two	Three	Four
Step	2.813	0.132	2.378	115.965
Cumulative	2.813	2.945	5.323	121.288

Table 4. Protocol 3 performance (in msecs)

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

than 102.7% faster than supersingular curves for encryption operations. Another promising optimization explored by Pirretti et al. [11] is the use of a random oracle construction [1, 3]. Such an approach is formally weaker than "standard" cryptographic models, but is often essential to making practical cryptosystems. As measured by Pirretti et al., this approach results in 395.9% faster encryption for supersingular and 408.4% for MNT curves.

4.3. Scalability

In order to support scalability to very large peer-to-peer systems, we consider protocol cost under replicated operation. In this evaluation, we assume that all server functions can be replicated, and that such replication leads to linear or near-linear speedup (a reasonable assumption).

We formulate the size of the community based on the protocol and optimizations as follows: Assume that a base server exchange takes k microseconds under an SS curve. Each construction has a optimization factor o that represents the protocol speedup factor (MNT=2.027, random oracle SS=4.959, and random oracle MNT=5.084). Further, assume an average occupancy of a user is s (in hours). Then, the supported community size $\mathcal{C}$ would be:

$$\mathcal{C} = \frac{10^6 * o}{k} * (s * 60^2) \quad (1)$$

Applying this formula to real environments, assume that users have an average occupancy in the peer-to-peer system of 2 hours (a conservative estimate), and that the node joining/rejoining is uniformly distributed in time. In this case, a system of two servers could support a user community of 626,000 and 619,000 users in protocols 2 and 3, and a system of only 50 servers could support over 15,800,000 and 15,600,000 users, respectively.

5. Discussion

ID-based cryptosystems have many advantages over certificate-based systems, such as simplification of key management. However, the operational requirements of ID-based cryptosystems present other challenges.

5.1. Key Escrow

One of the limitations of ID-based cryptography is an unavoidable presence of key escrow. To wit, a dependence exists on the trusted private key generator (PKG), which has full knowledge of all private keys in the system However, the server represents a single point of failure in the system: if the PKG is compromised, all private keys can be exposed.

Several schemes have been proposed to limit the effect of server compromise in ID-based cryptosystems. One such scheme uses multiple authorities to store and use the master key [2, 6], where no single authority ever possesses enough information to autonomously generate a private key. However, these solutions can add significant complexity to the system, e.g., complex failure modes, required additional protocol exchanges, etc.

5.2. Key Revocation

ID-based schemes do not need to manage Certificate Revocation Lists (CRLs) or verify the validity of public keys through a certificate chain. It is, however, inherently difficult to support proper key revocation in the system when a node's public key is synonymous with its ID.

Key expiry can be incorporated in an ID-based system by including the current date or time as part of the public key, along with the node ID [2]. However, the validity period affects the security of the system; if the time period is too short, updating the corresponding private key may introduce unnecessary computation at the PKG. Conversely, longer time periods can result in more exposure to compromise.

5.3. Denial of Service Attacks

The PKG in an ID-based cryptosystem may be attacked by sending a flood of forged or spoofed requests, overwhelming it with false requests for private keys. As shown in section 4, this key generation is computationally expensive, and a flood of false requests may result in the PKG ceasing to meaningfully function.

Possible solutions to these *Denial of Service* (DoS) attacks include implementing load balancing [7] or computational puzzles [13]. Ultimately, server resources are finite and achieving resilience to thousands or millions of malicious hosts is, to say the least, challenging. Defending against these attacks is beyond the scope of this paper.

6. Related Work

Douceur [9] identifies *Sybil attacks* as adversaries simultaneously obtaining many pseudo-identities in P2P systems. He suggests methods for imposing computational cost on creating an identity and system conditions to mitigate the attack. However, Douceur limits much of his discussion to the attack, and it is not clear how one would implement these approaches in P2P overlay networks.

In addition to a centralized authority, Castro et al. [4] suggest either charging money for certificates or binding node IDs to real-world identities to mitigate the Sybil attack. While this can ensure that node IDs are unique and, to some extent, moderate the rate at which node IDs can be obtained, it is often impractical to require that all nodes spend money or prove their real-world identity in P2P systems.

Srivatsa and Liu [17] espouse a variant of the traditional approach. In this, the bootstrap node assigns a random identifier and issues an associated certificate with a short lifetime. This can guarantee unique node ID assignment and also control the number of node IDs that are generated in the system. However, it can be cumbersome for all nodes to obtain and update a certificate.

A variety of cryptographic puzzle mechanisms have been proposed to address Sybil attacks. Rowaihy et al. [13] present an admission control system using a hierarchy of participating peers and a chain of puzzles. However, it is limited by a complex structure and requires a potentially large number of exchanges with varying servers to obtain a single ID.

7. Conclusion

In this paper we have considered the use of identity-based cryptography to assist in the security and performance critical assignment of user identities in peer-to-peer systems. We developed three protocols representing diverse trust models and performance profiles based on identity-based cryptography. Our evaluation of the performance of these protocols shows that their costs vary widely by model and type of cryptography used. We further show that systems using these protocols can scale to massive P2P networks through the proper use of cryptography and server replication.

Peer-to-peer systems often face conflicting requirements for autonomy, robustness, and security. These systems fill an important niche by providing highly-available, massively-distributed storage. However, their continued growth is dependent on the technical community's ability to introduce further infrastructure to secure the media. This work and others like it will solve the challenges of this media by exploiting emerging technologies such as identity-based cryptography.

References

[1] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pages 62–73, 1993.

[2] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer-Verlag, 2001.

[3] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[4] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of OSDI 2002*, Boston, MA, Dec. 2002.

[5] J. C. Cha and J. H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Public Key Cryptography*, pages 18–30, 2003.

[6] L. Chen, K. Harrison, N. Smart, and D. Soldera. Applications of multiple trust authorities in pairing based cryptosystems. In *Proccedings of the Infrastructure Security Conference 2002*, volume LNCS 2437, pages 260–275, 2002.

[7] N. Daswani and H. Garcia-Molina. Query-flood DoS attacks in Gnutella. In *Proceedings of ACM CCS'02*, pages 181–192, Washington, DC, 2002.

[8] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[9] J. Douceur. The Sybil attack. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*, Cambridge, MA, March 2002.

[10] B. Lynn. PBC library. http://rooster.stanford.edu/~ben/pbc/, 2006.

[11] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure Attribute-Based Systems. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, November 2006.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, , and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM 2001*, pages 161–172, San Diego, CA, 2001.

[13] H. Rowaihy, W. Enck, P. McDaniel, and T. La Porta. Limiting Sybil attacks in structured peer-to-peer networks. Technical Report NAS-TR-0017-2005, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, July 2005.

[14] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of Middleware*, pages 329–350, Heidelberg, Germany, 2001.

[15] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 47–53. Springer-Verlag New York, Inc., 1985.

[16] A. Singh, M. Castro, P. Druschel, and A. Rowstron. Defending against eclipse attacks on overlay networks. In *Proceedings of ACM SIGOPS European Workshop*, Leuven, Belgium, 2004.

[17] M. Srivatsa and L. Liu. Vulnerabilities and security threats in structured overlay networks: A quantitative analysis. In *Proceedings of ACSAC 2004*, pages 252–261, Cambridge, MA, 2004.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of ACM SIGCOMM 2001*, pages 149–160, San Diego, CA, 2001.

[19] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.