Decentralized Trust in Distributed Networks: a Delegate -based Security Hardening Approach

Jabeom Gu, Jaehoon Nah, and Jongsoo Jang

Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea

Summary
Current peer-to-peer (P2P) networks are well defined in their communicating and collaborating mechanisms – search, share, and retrieve information – especially when there is a centralized management server. However, these operations provide no convenient security mechanisms for serverless networks. This paper examines a security hardening approach to limit or prevent identifier attacks between distributed P2P networking nodes without help from a centralized server.
This paper introduces the basic concept of security hardening and discusses how it can be realized in distributed P2P networks. This paper provides a brief review of a relevant work from the literature. It is a method of a peer to create a cryptographic 'trust binding' with a remote peer easily and instantly without any help of a centralized manager or server, with which the first peer can interpret the cryptographic binding as its trust upon the remote peer. Because the scheme combines peers identifier to the generated trust binding, an attacker cannot do much harm (like misrouting, corrupting, or dropping communication data transferred between the first and the second peer) to the peers, without breaking the binding. Also presented is a detailed analytic study of the security hardening approach from which the strength of the scheme is discussed. Our analysis shows that once implemented the security hardening can be an important countermeasure against various identifier related attacks in distributed P2P networks.

Key words:
Peer-to-peer, security, identifier authentication, unmanaged networks, security hardening

1. Introduction

One common approach to handle system and network security problems is using hardening strategy. Hardening may involve, for example, the process of evaluating, auditing, and/or some countermeasures to secure system and network resources. In general, security hardening is a series of acts like configuring systems and developing related applications in a way to maximize (system) security to prevent or limit various attacks [1, 2, 3, 4]. Authorized managers shall conduct a variety of techniques for hardening and fortification whenever needed. Therefore, security hardening is generally considered to be appropriate for a controlled domain such as enterprise network.

In large-scale distributed networks, however, the administrative and enforcement difficulties/costs involved, not to mention the performance liabilities, make widespread implementation of security hardening completely impractical. Furthermore, given the fact that a significant fraction of end users in public domain are not legitimate nor trustworthy in some aspects, the network be vulnerable to so called "insider attacks" such as disruption, cheating, man-in-the-middle, and denial-of-service caused by those users [5].
Motivated by this perception, we explore the concept of security hardening for trust management in large, distributed networks especially when there is no centralized authority function. The trust is closely related to the authenticity of the binding between a public key and a user: "*a user can be said to trust a second user when the first user can verify the authenticity of a public key that is claimed to be held by the second user.*" This type of trust, which is also known as **key authentication**, is the main benefit the PKI (Public Key Infrastructure) would provide in a controlled domain. However, because the importance of the true identity of a user in large scale distributed networks, such as P2P (peer-to-peer) network, seems to be diminishing, we narrowed the scope down to authenticity of the binding between a public key and an identifier of a node: "*a node can be said to trust a second node when the first node can verify the authenticity of a public key that is claimed to be held by the second node*." The new type of trust is referred as **identifier authentication**.
The identifier authentication is gaining significant importance in distributed networks because many attacks like misrouting, corrupting, or dropping communication messages and the routing information become much easier as a result of a successful identifier forgery [6, 7]. In general, the blanket term "Sybil attack" [6] is used to encompass the process of generating (forging) large number of identifiers and/or the resulting attacks caused by exploiting those identifiers.
Some countermeasures for the Sybil attack would be 1) limiting number of identifiers that a node can generate and 2) making the cost of identifier generation function relatively high. As discussed in [6], however, limiting the number of identifiers that a single node can generate is not easy in distributed, authority-less networks. The general assumption in network security – an attacker may have

Manuscript received October 5, 2007
Manuscript revised October 20, 2007

unlimited resources compared to ordinary users – also disables realization of the second method. When there are indications of a possibility of the Sybil attack, it is difficult to validate trustworthiness of a claimed identifier. These attacks are easy to be exploited because of the open nature of the distributed networks.

In response to this, several approaches are studied in the literature [8, 9, 10, 11, 12, 13]. This paper investigates one such scheme, namely, security hardening by trust binding, originally proposed by J. Gu et al. [13]. The paper introduces a novel trust management scheme, in which trust between two nodes is achieved by carefully choosing delegate node(s) within the network and exchanging cryptographically protected key material to create one or more indirect trust binding(s) through the chosen delegate(s). The exchange result is referred to as *trust binding* because a node can create a virtual binding with other node by combining identifier and public key of that node to the exchanged key material, which in turn makes an attacker very difficult to harm (such as misrouting, corrupting, or dropping communication data) the communication between the two nodes without breaking the cryptographic protection. The exchange result is also referred to as *indirect* because the scheme uses carefully chosen node(s) as delegate(s) of the trust binding. The protocol or the underlying protection mechanism to exchange the key material is very simple, but we are focusing on that such simple transaction is enough to mitigate identifier attacks and, thus, to ensure trustiness of involved nodes.

To achieve the trustiness (i.e., identifier authentication) the proposed scheme combines nodes identifier and public key to the indirect trust binding. Therefore, once the trust binding between two remote nodes is generated successfully, it can be interpreted that the identifiers are not forged or there is no on-going identifier related attacks between the two nodes. In other words, the scheme is designed to prevent identifier attacks by forcing the attacker to break the indirect trust binding.

The remainder of this paper is organized as follows. Section 2 provides brief discussion on trust management in distributed networks. Section 3 presents basic assumptions. Section 4 presents review of security hardening approach proposed by J. Gu et al. [13]. Section 5 presents security analysis and discusses the strength of security hardening. Section 6 concludes the paper.

2. Trust management in distributed networks

The trust issue "*whether I can trust the remote peer with which I am talking over the Internet*" is commonly interpreted in general *controlled networks* as the trust for the name (i.e., true identity) or role (i.e., what s/he do) of the peer. However, in large scale distributed networks the importance of the name of a user seems to be diminishing [5]. In P2P file sharing networks such as Gnutella [14], for example, users are not actually interested the true identity of a peer who is sending or requesting a file.

The PKI (Public Key Infrastructure) based trust management is a good solution generally employed in corporation-like *controlled domains*. However, some large-scale, distributed networks like P2P (Peer-to-Peer) is too *open*, too *heterogeneous*, and too *big* for the PKI[1].

3. Problem setting

We consider a distributed, unmanaged P2P network with large number of peer nodes. In the following discussion, we assume that a peer node (or an **initiator**) tries to communicate with a remote peer (or a **target**). When a P2P network is not managed by some administrative body, it naturally means that "*peers can join and leave the network without any restriction and at any time they want*." Because no management server is available, each peer should advertise its information to other peers frequently. Therefore each peer has a different '*view*' of the P2P network from other peers. Furthermore, the view is time varying. When a peer is connected to the P2P network, it has empty view. It must collect advertised information of other peers to build-up its own view.

We assume:

1) A peer can freely generate as many identifiers as one needs;
2) No centralized management authority;
3) NAT or DHCP based transitory IP address;
4) Two nodes represented by two different IP addresses are treated as different nodes;
5) Peers can only see part of the network.

An attacker can generate many identifiers to achieve

1) To look like someone else (disguise);
2) To deny identifier related management functions like DHT, in which identifiers are used to locate and lookup;
3) To repudiate ones action(s) under a certain identifier;
4) To forge voting systems.

The proposed scheme does not address:

1) A peer who's real identity is Alice, but acting as someone else. (Anonymous) We cannot distinguish one peer from another.

The proposed scheme address:

1) Two nodes who are using same identifier are distinguishable to verifier;

[1] Note that the specific P2P based VoIP provided by the Skype employed self-generated public key based mutual trust between peers. In such case, centralized server manages the mapping between peer identity and his/her public key, minimizing possibility of identity theft.

2) If one node uses a specific identifier, another node cannot use that same identifier, which can be detected by 1).

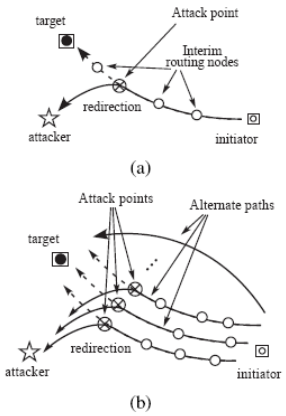


Fig. 1: Example of the target based attacks in multi-hop relay based communications: (a) when there is only one path, and (b) when there are multiple paths.

Suppose that an attacker has forged enough number of identifiers and placed several 'slave' nodes along the routing path from an initiator to a target. This is illustrated in Fig. 1. In the figure, crossed circles indicate adequate attack points. In the example scenario of Fig. 1(a), an attacker is trying to re-route all packets addressed to the target. That is, if those slaves can re-route packets from the initiator to the target, then the attack is considered successful. In the second example in Fig. 1(b), when there are several alternate routing paths from the initiator to the target, the attack is considered successful only when the attacker can forge all paths.

4. Security hardening

A simple metaphor that is commonly found in everyday Internet service can be used to present the underlying concept of a security hardening approach presented in [13]. Consider that Bob is trying to register himself to a Web server through the Internet. The scheme, which we call "registration by e-mail confirmation," states as: "*to confirm your registration, we will send an e-mail to the e-mail address you have submitted. You must reply to the e-mail within 12 hours to complete your registration. If you do not, the registration information will be deleted from the file*."

In this example, Bob is the target whose identity needs to be proved; and the Web server who sends a confirmation message is the initiator. The overall procedure is presented in Algorithm 1.

This simple metaphor captures a number of important characteristics of the security hardening, each of which is de-scribed separately below.

- **Moderate trust on the user**: Trust of the above procedure dependent (1) primarily on the key material can only be accessible by the one who has access to the e-mail address, namely *bob@abc.com*, and (2) partly on the fact that Bob is already a registered user of the e-mail server. If Bob cannot access the e-mail or fails to respond within due time, the on-line registration fails.
- **Deduced trust**: If the Web server has little or no *trust* on the e-mail server, so it does on Bob. Therefore the trust is deduced from the chance by which the e-mail server will evoke Bob and deliver □ successfully. The 'trust' in this case means that the e-mail server is no being attacked with high probability, and it does not mean that the e-mail server and the Web server has any preconfigured credential (i.e., shared secret or public key certificates issued by a mutually trusted third party).

Algorithm 1 Example flow of an on-line registration scheme

S1 Bob fills out an on-line registration form and posts it to the web server.

S2 The web server accepts the request and sends, some time later (t_0), a confirmation e-mail with key material ϵ to Bob's e-mail address, say *bob@abc.com*.

S3 The Bob's e-mail server evokes him when the message arrives.

S4 Bob manipulates ϵ to generate a proper response.

S5 To complete, Bob should continue the registration procedure within due time using generated response.

- **Empirical tight binding**: Several aspects of the registration procedure create tight bindings between the Web server and the e-mail server; and between the e-mail server and Bob such that they are *enough* to pre-vent possible attacks. That is, if an attacker is trying to forge the registration procedure from Bobs side, he must break the binding between Bob and the e-mail server; if he is trying to attack from the side of the Web server, he must break the binding between the Web server and the e-mail server. Breaking the communication path from Bob to the Web server is not enough. In the metaphor, it is also not defined *when* and *how* the Web server will solicit a proxy from the e-mail server, which decrease or limit the possibility an attacker can try some predefined attack.

4.1 Security hardening using trust binding

The security hardening proposed in [13] uses instantly generated moderate *trust binding* through a randomly

selected delegate[2] (as shown in Fig. 2(a)) or through multiple random delegates (as shown in Fig. 2(c)). If the delegate(s) are selected randomly enough, it will make an attacker very hard to predict the exact identifier of the delegate. By combining identifier of peer into the moderate trust link, an attacker cannot do much harm (like misrouting, corrupting, or dropping communication data) to the initiator or the target. Therefore, if an attacker is trying to launch any identifier related attack, it must break the moderate trust binding(s).

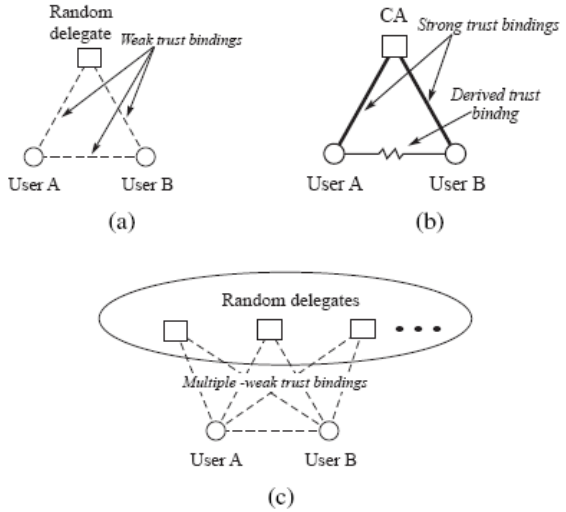


Fig. 2: Comparison of *moderate* trust binding scheme with the PKI trust model. (a) The trust binding scheme generates three moderate trust bindings between three tiers <*initiator - delegate – target*>. (b) In PKI, trust between two peers is derived from the two *strong trust bindings* between CA and each peer. (c) Strengthening trust binding by repeating the scheme several times using multiple delegates.

Based on this, authors of [13] proposed an identifier verification protocol. That is once the trust binding between two remote peers can successfully generated through a randomly selected delegate, it can be interpreted that the identifier of the peers are not forged or there is no on-going identifier related attacks between the two peers. Strength of the scheme will be discussed in Section 5.

Fig 2 shows a comparison of the trust binding scheme with that of the Public Key Infrastructure (PKI). The PKI is a collection of technical and organizational facilities that provides trusted third party to attest the **key authentication problem** – the identity of the public key holder should be bound to the public key of that holder. In PKI, a trustworthy third party (called certificate authority or certification authority (CA)) is arranged to publish trust materials, called certificates, which can be used to validate the binding between a user and his or her public key. Thus trust between users and the CA is a strong one and the trust between users is a derivation of the strong trust. This is illustrated in Fig. 2(b).

Considering this, trust binding scheme in [13] achieves:

- **Blinding attacker**: first and most evident from the scenario in Fig. 1(a) is that fixed nodes can easily be a tar-get of an attack. One possible solution is to blinding so that an attacker cannot premeditate an action. This is realized in [13] by ensuring randomness in deter-mining validity of peer identifier. Suppose in the Bobs example in the previous section that Bob has several e-mail addresses, namely *bob@def.com*, *bob@ghi.com*, etc. While the attacker does not know which one will be used, it shall try to break all the bindings. Similar concept is realized in [13] by selecting several delegates randomly from a peers view.
- **Moderate trust**: authors of [13] contend that a restricting approach, even a weak one, is enough for P2P overlay networks to make attacker a demanding job. Actually in many distributed P2P open environment, using authentication mechanism like PKI is too much of a luxury, not to mention technically infeasible. The trust binding achieves high level of trust by introducing a randomly selected delegate and binding three tiers <initiator-delegate-target> using moderate, efficient mechanism based on identity-based cryptography (IBC) [15, 16, 17, 18, 19].
- **Decreasing possibility of successful attack**: a simple way to do this is to use parallel redundancy assuming that attack events against peers are mutually independent. As illustrated in Fig. 1(b), repeated application of the trust binding scheme will naturally provide this feature.

4.2 Review of protocol

(Setup Phase)
P0 i : acquire target advertisement $\langle \mathtt{ID_t} \mid \mathtt{IP_t} \mid K_t^u \rangle$
(First Phase)
P1 $i \longrightarrow t$: $\langle \mathtt{ID_i} \mid \mathtt{ID_t} \mid N_i \rangle$
P2 t : $aid_t = H(\mathtt{ID_t} \mid K_t^u \mid N_i \mid N_t \mid \mathtt{IP_t})$
P3 $i \longleftarrow t$: $\langle aid_t \mid K_t^u \mid E_{aid_i}(N_t) \mid S_{K_t^r}(aid_t) \rangle$
(Second Phase)
P4 i : $N_t \leftarrow D_{aid_i}(E_{aid_i}(N_t)), \quad \epsilon = E_{aid_t}(N_v \mid N_t)$
P5 $i \rightsquigarrow d$: $\langle E_{aid_d}(aid_t \mid aid_i \mid \epsilon) \mid S_{aid_i}(\epsilon) \rangle$
P6 $d \rightsquigarrow t$: $\langle E_{aid_t}(aid_t \mid aid_d \mid \epsilon) \mid S_{aid_d}(\epsilon) \rangle$
P7 t : $N_v \leftarrow D_{aid_t}(\epsilon), \quad pr = H(aid_t \mid aid_i \mid N_v)$
P8 $i \longleftarrow t$: $\langle E_{aid_i}(pr) \mid S_{aid_t}(pr) \rangle$
P9 i : check validity of proof material

Fig. 3: Identifier security protocol [13]

The protocol steps are shown in Figure 3. Among the steps, the message flow direction between three peers – initiator, target, and random delegate – is illustrated in Figure 4. The

[2] The peers in one's view can be considered good candidate for this purpose.

protocol of the proposed identifier security mechanism includes sending an initialization request message (P1), sending a response message (P3), sending an indirect binding request message (P5), sending a forwarded indirect binding request message (P6), and sending an indirect binding confirm message (P8).

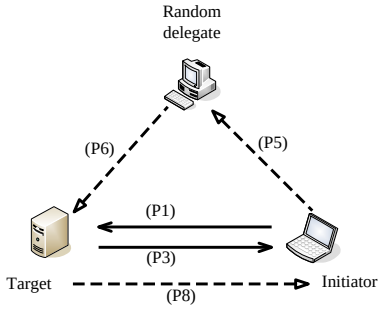


Fig. 4: Message flow between three peers: initiator, target, and random delegate

In the first phase (P1 — P3), the initiator sends a request for identifier verification protocol to the target. The nonce N_i included in the request message ensures that the identifier verification is fresh. Upon receiving the request message, the target generates new auxiliary identifier aid_t. The target sends response message back to the initiator. Because the N_t needs to be available only to the initiator, it is encrypted using initiator's public auxiliary identifier (aid_i). The procedure (P3) provides ownership of the RSA private key K_t^r by combining public key signature (i.e., $S_{K_t^r}(aid_t)$) into the message, which can be used later for session key establishment.

In the second phase (P4 — P8), the initiator generates a key material ε using the response from the target, and delegates the proof mechanism to a randomly selected node d. The key material is generated using

$$\varepsilon = E_{aid_t}(N_v \mid N_t),$$

where N_v and N_t are nonces generated by initiator and target respectively. In other words, the ε is an encrypted value of the two nonces using the target identifier aid_t as the key. Therefore, secrecy of the IBC guarantees that only the target can see the nonce N_v.

The delegation message is signed using $S_{sid_i}(\varepsilon)$ so that the delegate d can verify the authenticity of the message. On receiving the delegation message, the delegate simply forwards ε to the target. To do this, the delegate reproduces its own delegate message and appends its own signature.

When the target t receives ε from the delegate, it can decrypt it and prepare a proof material pr. The target sends the pr along with its signature to the initiator. The initiator checks the signature and the hashed proof pr. If the hash matches, the identity security mechanism completes.

As discussed in the previous subsection, the strength of the random visitor scheme partly relies on the random selection of the delegate from the delegate space such that an attacker faces the difficulties of placing some forged identifiers near the randomly selected delegate in due time. We evaluate effectiveness of these achievements for identifier security hardening in the numerical studies that follow.

5. Analysis of hardening approach

In this section, we present numerical investigations into the strength of the security hardening approach. In particular, using a simplified model of the identifier and *view*, we quantify how the hardening achieves desired level of security.

The trust binding in [13] is realized using IBC scheme. But without the help of trusted third party as a private key generator (PKG), there is the same key authentication problem. This is the reason why the trust binding scheme is called “moderate security.” However, we should note that the strength of the scheme does not rely on the authenticity of the IBC key held by each peer node. Instead the strength of the scheme depends on the basis that the collection of such moderately secure binding is enough to make an attacker hard to perform an identifier forgery attack.

Let Δ denote the random delegate space. In general cases, Δ is equal to the *view* of a peer. If an attacker has attacked sufficiently large set of identifiers that belong to Δ, there is the possibility that the initiator may select, as a random delegate, a node that has already been compromised by the attacker. We refer to this situation as a *collision*. In this case, the attacker can control (or break) the binding between the initiator and the delegate; or the binding between the delegate and the target. Even in such situation, however, the attacker can not cause any other malicious harm but to deny being a proxy of the identifier verification procedure and drop the delegation request message. The initiator will eventually time out and try another delegate.

Assuming that an attacker can compromise randomly selected ω nodes from the delegate space Δ, and assuming that the initiator can only see another random candidate list of ω nodes from the same delegate space, a rough estimation of the collision probability P_c that at least one

of the attackers selection belongs to the initiators candidate list is

$$P_c = Pr(\text{Select at least one candidate list}) \quad (1)$$

We found that a technique [20, 21] for analyzing solution for key pre-distribution in sensor networks can also be applied, with a little modification, to find P_c as

$$\begin{aligned} P_c &= 1 - \frac{\binom{|\mathcal{D}|}{\tau}\binom{|\mathcal{D}|-\tau}{\tau}}{\binom{|\mathcal{D}|}{\tau}^2} \\ &= 1 - \frac{((|\mathcal{D}|-\tau)!)^2}{(|\mathcal{D}|-\tau)!(|\mathcal{D}|)!}. \end{aligned} \quad (2)$$

Fig. 5 shows several plots of this function. The figure indicates for a network with |Δ| = 100000 delegates, even when an attacker can compromise 100 nodes, the attacker the probability is 009. This points out that randomly selecting a delegate from a space still has low collision probability if the space is large enough. Fig. 6 shows the desired size of the random delegate space |Δ| to keep the probability of collision P_c low enough (around 0.1, 0.2, and 0.3, respectively).

In order to obtain desired possibility of successful identifier verification, the trust binding scheme should be performed multiple times. Let ℓ denote the number of successive trials and *p(i)* the probability that there is exactly *i* successive collisions. That is,

$$p(i) = \frac{\binom{|\mathcal{D}|}{i}\binom{|\mathcal{D}|-i}{2(\tau-i)}\binom{2(\tau-i)}{\tau-i}}{\binom{|\mathcal{D}|}{\tau}^2}. \quad (3)$$

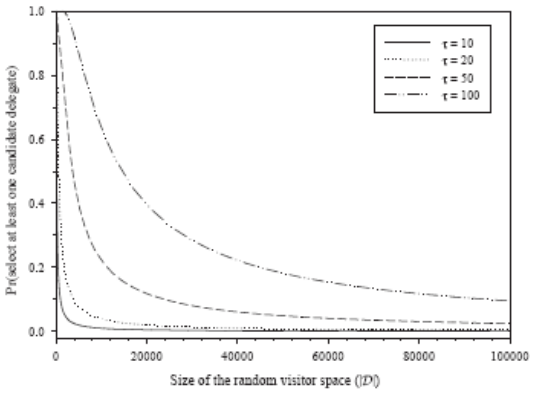


Fig. 5: Probability of selecting at least one delegate that has already compromised assuming that an attacker can compromise ˷ nodes from the total delegate space |Δ|.

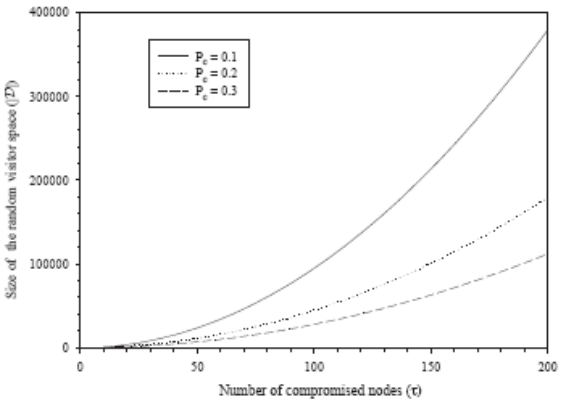


Fig.6: Desired size of the random delegate space when an attacker can compromise nodes.

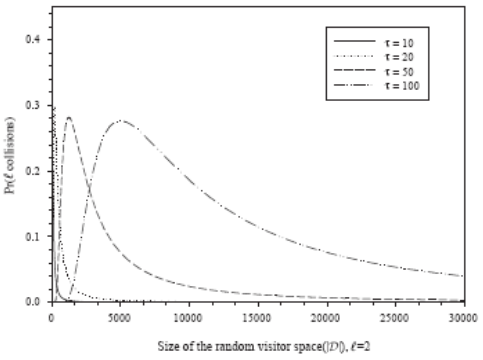
Note that the P_c is a special case of *1-p(0)*. When an initiator performs ℓ verifications, the attacker can succeed only when it forged all ℓ delegates. Thus the probability of ℓ collision is

$$\begin{aligned} P_{c(\ell)} &= p(\ell) \\ &= \frac{\binom{|\mathcal{D}|}{\ell}\binom{|\mathcal{D}|-\ell}{2(\tau-\ell)}\binom{2(\tau-\ell)}{\tau-\ell}}{\binom{|\mathcal{D}|}{\tau}^2}. \end{aligned} \quad (4)$$

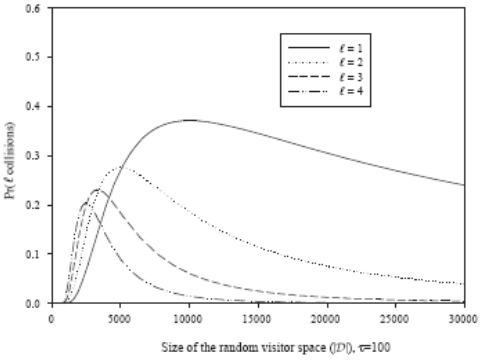
Evaluation of equation (4) is shown in Fig. 7. In Fig. 7(a), we compared the probability of ℓ-collisions $P_{c(\ell)}$ for a fixed ℓ = 2 while ˷ is set to 10, 20, 50, and 100. Fig. 7(b) compares $P_{c(\ell)}$ for a fixed ˷ = 100 while ℓ is varying between 1 and 4. Compared to the Fig. 5, these diagrams show that repeating trust binding scheme is still resilient for 100 identifier forgeries from relatively small delegate space Δ.

6. Conclusion

In this paper, we have reviewed identifier related attacks and presented an analytic study of the identifier security hardening approach called trust binding scheme. An interesting observation was that the scheme bases its strength on the fact that the collection of moderately secure binding is enough to make an attacker hard to perform an identifier forgery attack. The analysis revealed that randomly selecting a delegate from a space still has low collision probability if the space is large enough. Furthermore, the analysis result showed if the scheme is repeated several times, typically three or four times, the P2P network is resilient for large identifier forgery attacks from relatively small delegate space.



(a)



(b)

Fig.7: Probability of ℓ collisions. (a) for various values of τ when $\ell = 2$. (b) for various values of ℓ when $\tau = 100$.

Security hardening can be realized in many different ways. The trust binding scheme investigated in this paper is one such approach to be used in P2P networks. We expect that the analysis results presented in this paper be used as design guidelines to achieve trustworthiness without the help of centralized manager.

The security level that can achieved through implementing the security hardening mechanism may be increased if there are many peers (more than several thousand) in the network. However, if the network is going to be managed by a centralized authority or if there are very few peers in the network, the security achievement through implementing the proposed mechanism may be reduced or minimized. Therefore, using proposed mechanism in such networks as a base identifier security mechanism is not recommended.

Acknowledgment

This work was supported by the IT R&D program of MIC/IITA [2005-S-090-03, Development of P2P Network Security Technology based on Wired/Wireless IPv6 Network].

References

[1] D. Szajda, B. Lawson, and J. Owen, “Hardening functions for large scale distributed computations,” in Proceedings of Symposium on Security and Privacy 2003, pp. 216-224, 2003.

[2] S. Noel, S. Jajodia, B. O’Berry, and M. Jacobs, “Efficient minimum-cost network hardening via exploit dependency graphs,” in Proceedings of 19th Annual Computer Security Applications Conference, pp. 86-95, 2003.

[3] K. Kyamakya, K. Jobman, and M. Meincke, “Security and survivability of distributed systems: an overview,” in Proceedings of 21st Century Military Communications Conference (MILCOM 2000), vol. 1, pp. 449-454, 2000.

[4] T. Jackson and M. Wilikens, “Survivability of networked information systems and infrastructures: First deliverable of an explanatory study,” European commission special report, JCR/ISIS/STA/DAS/Projects/Survivability/Study, 1998.

[5] M. Bursell, “Security and trust in p2p systems,” in *Peer-to-Peer Computing: The Evolution of a Disruptive Technology*, R. Subramanian and B. D. Goodman, Eds. Idea Group Publishing, pp. 145-165, 2005.

[6] J. Douceur, “The sybil attack,” in *Proc. 1st International Peer-To-Peer Systems Workshop (IPTPS)*, Cambridge, MA, pp. 25 1260, 2002.

[7] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, “Secure routing for structured peer-to-peer overlay networks,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 299-314, 2002.

[8] L. Ganesh and B. Y. Zhao, “Identity theft protection in structured overlays,” in *Proc. of 1st Workshop on Secure Network Protocols (NPSec)*, Boston, MA, 2005.

[9] H. Rowaihy, W. Enck, P. McDaniel, and T. L. Porta, “Limiting sybil attacks in structured peer-to-peer networks,” Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, Tech. Rep. NAS-TR-0017-2005, July 2005.

[10] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. Anderson, “Sybil-resistant DHT routing,” in *Proc. 10th European Symposium On Research In Computer Security*, 2005.

[11] J. Dinger and H. Hartenstein, “Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration,” in *Proc. First International Conference on Availability, Reliability and Security (ARES)*, pp. 756-763, 2006.

[12] A. Singh, M. Castro, A. Rowstron, and P. Druschel, “Defending against eclipse attacks on overlay networks,” in *Proc. 11th ACM SIGOPS European Workshop*, Leuven, Belgium, 2004.

[13] J. Gu, J. Nah, C. Chae, J. Lee, and J. Jang, “Random visitor: a defense against identity attacks in overlay networks,” in *Proc. 7th International Workshop on Information Security Applications (WISA)*, Jeju, Korea, pp. 601-615, 2006.

[14] G. Kan, “Gnutella,” in *Peer-to-Peer: Harnessing the Benet of a Disruptive Technology*, A. Oram, Ed. OReilly & Associates, Inc., Sebastopol, CA, 2001.

[15] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *Proc. CRYPTO 84 on Advances in Cryptology,* Santa Barbara, CA: Springer-Verlag New York, Inc., pp. 47-53, 1985.

[16] E. Okamoto and K. Tanaka, "Key distribution system based on identification information," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 481-485, 1989.
[17] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Proc. 8th IMA International Conference on Cryptography and Coding*, Springer-Verlag, pp. 360-363, 2001.
[18] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAMJ. Comput.*, vol. 32, no. 3, pp. 586-615, 2003.
[19] L. Martin, "Identity-based encryption: A closer look," *ISSA Journal*, pp. 22-24, 2005.
[20] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228-258, 2005.
[21] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proc. IEEE Symposium on Security and Privacy 2003.

Jabeom Gu received the M.S. and Ph.D. degrees in Electrical Engineering from Chung-Ang University in 2002 and 2006, respectively. He is with Electronics and Telecommunications Research Institute, Korea. His research interest includes distributed network security, peer-to-peer network, overlay multicasting, wireless network security, and IPTV security.

Jaehoon Nah received the M.S. degree in Computer Engineering from Chung-Ang University in 1987. He received the Ph.D. degree in Electronic and Infomation Engineering from Hankuk University of Foreign Studies in 2005. He is a principal research engineer and a team leader in Division of Information Security in Electronics and Telecommunications Research Institute, Korea. His research interest includes distributed network security, peer-to-peer network, overlay multicasting, and IPTV security.

Jongsoo Jang received the B.S and M.S degrees in Electronics Engineering from Kyungpook National University in 1984 and 1986, respectively. He received his Ph. D degree in Computer Engineering from Chungbuk National University in 2000. Since 1989, he has been working with ETRI, Daejeon, Korea and now is the Director of Applied Security Group.