AREX: An Adaptive System for Secure Resource Access

Brent Lagesse, Mohan Kumar and Matthew Wright
Department of Computer Science Engineering
University of Texas at Arlington
{Brent.Lagesse, mkumar, mwright}@uta.edu

Abstract

In open environments, such as mobile peer-to-peer systems, participants may need to access resources from users they don't know. One of the most critical security issues this raises is that the resources accessed could be faulty, thereby wasting the requester's time and energy and possibly causing damage to her system. A common approach to mitigating the problem involves reputation mechanisms; however, since reputation relies on cooperation, a reputation mechanism's effectiveness can be significantly diminished in hostile environments. Reputation systems also require substantial communication between peers, making them vulnerable to errors caused by intermittent connectivity and attackers selectively disrupting message delivery. The communication also results in message overhead, which is potentially expensive for mobile environments in terms of energy costs. In this paper, we present AREX, a mechanism designed to provide security for peers in hostile and uncertain environments, which are common in mobile P2P systems. AREX features an adaptive exploration strategy that increases the system's utility for benign peers and decreases the system's utility for malicious peers. AREX reduces vulnerabilities and energy costs by operating without communication between peers. Through simulation, we demonstrate AREX's ability to reduce energy costs, protect benign peers, and diminish malicious peers' motivation to attack in a variety of hostile environments.

1. Introduction

Mobile P2P systems will enable users to share information and resources in a variety of open environments without the need for pre-existing infrastructure. They can help facilitate the work of first responders, police, and military, as well as providing useful services wherever people gather. While enabling fast and easy interaction and sharing, however, the openness of such systems also makes them relatively easy for an attacker to exploit. In this paper, we focus on an attack in which malicious peers provide faulty resources — resources that are not what was requested or include a malicious payload. We define the term resources broadly to mean any service or data that the user may request. When requests are met with faulty results, the requesting peer suffers wasted time, wasted energy, and potential harm to its system.

In this paper, we propose Adaptive Resource Exploration (AREX), a novel mechanism to improve the security of resource accesses in P2P systems. AREX promotes secure resource accesses by using *resource exploration*, in which peers perform extra resource accesses to test the reliability of other members of the system. These exploratory accesses enable users to find reliable peers that are more likely to respond when the user needs real resources. A distinctive feature of the proposed mechanism is its ability to decrease the energy consumption of a peer in a hostile environment, which is crucial to mobile P2P systems. A preliminary version of this paper, introducing the resource exploration strategy, was presented at the 5th IEEE International Workshop on Mobile Peer-to-Peer Computing [16]. In this enhanced paper we present a refined and more comprehensive scheme that overcomes the challenges inherent in game theoretic and utility-based approaches. We also include results of extensive simulation studies that show AREX's energy conservation and benign peer protection abilities.

We evaluate AREX in the following three environment types:

- Uncertain/Malicious systems
- Systems with intermittent connectivity
- Systems with peers that are sensitive to attack

Some reputation mechanisms rely on the assumption that a subset of the peers is *pre-trusted* [2]. However, it is not always possible to identify such peers in uncertain and potentially malicious systems. For example, in a file-sharing application in an urban setting, users may continuously come and go, making it difficult to identify peers that can be pre-trusted. In hostile environments, pre-trusted peers could be

captured and corrupted, so it can be dangerous to give a few nodes power to manipulate trust in the system. Without pre-trusted peers, there is no guarantee that reputation values provided by any peer are legitimate. Consequently, reputation mechanisms that accumulate the preferences of the majority of peers to calculate reputations will fail to provide protection for benign peers when they are in the minority.

Since reputation mechanisms are cooperative, they require communication among peers. In systems where consistent connectivity cannot be assumed, such as a mobile P2P system, a reputation mechanism will degrade in effectiveness when the portion of the system available to communicate with at any given time decreases [2]. The reduced ability to acquire reputation information may result in less reliable reputation results. The unreliability of reputation in these cases is caused because changes in reputation values that otherwise would have propagated quickly through the system now take longer, so peers make decisions based on degraded reputation information. Additionally, attackers may take advantage of wireless, peer-routed communication to selectively disrupt communications, thereby choosing which reputation information gets received.

One of the inadequacies of reputation-based security in P2P systems is the requirement of prior experience to make decisions [6]. As a result, peers are vulnerable against attacks when they enter the network. In a foreign system with no known trusted peers, an entering peer is vulnerable to attack as it has no means to determine the trustworthiness of any other peers in the system. This fact can be exploited by an individual malicious peer or by a set of collaborating peers. Furthermore, a peer can initially behave benignly, be recognized as such, and then act maliciously (either intentionally or due to being compromised). These attacks are especially dangerous for a peer that is sensitive to attacks (or to a particular type of attack) and reputation does little to prevent such attacks.

In this paper, we present novel aspects of the AREX solution that improve resource access security in three critical ways:

- Handling benign, but faulty peers
- Coping with differences in strategies within the system
- Achieving Nash equilibrium with no *a priori* knowledge

AREX achieves these properties by seeking to maximize the peer's utility while lowering the utility of providing malicious results. AREX quickly adapts to the environment it is in by randomly exploring and exploiting the resources available to the peer. Through this adaption, AREX is able to manage the trade off between exploration and exploitation to provide the peer running AREX with improved utility over naive approaches. Furthermore, in simulated environments we test the system against strategic attackers and validate our algorithms for several attacker models and varying types of benign peers.

U_{ben}	Utility for a Benign Peer
U_{mal}	Utility for a Malicious Peer
U	Total Utility
B	Total Benefit
C	Total Cost
B_{ben}	Benign Benefit
B_{acc}	Access Benefit
B_{mal}	Malicious Benefit
C_{ben}	Benign Cost
C_{mal}	Malicious Cost
C_{vic}	Cost from being a Victim

Table 1. Notations frequently used in this paper

2. Resource Exploration

We provide a brief explanation of resource exploration in this section. For simplicity in our analysis, we examine purely benign versus purely malicious peers. The details of the utility model are presented in [16]. Purely benign peers, as modeled by Equation 1, only gain utility from successful transactions and can be modeled with the parameters Benign Benefit, Benign Cost, and Victim Cost. Malicious peers, as modeled by Equation 2 only gain utility from successfully attacking other peers and can be modeled by using the parameters Malicious Benefit, Benign Cost, Discovery Cost and Malicious Cost.

$$U_{ben} = B_{acc} - (C_{ben} + C_{vic}) \quad (1)$$

$$U_{mal} = B_{mal} - (C_{ben} + C_{disc} + C_{mal}) \quad (2)$$

The main idea of the resource exploration is to send out exploratory requests in addition to real requests. This process is detailed in Algorithm 1. These exploratory messages are designed to reveal the nature of the peers resulting in an increase in utility for the requesting peer and a decrease in utility for attacking peers. Peers will incur a cost by sending the exploratory messages in terms of a greater amount of Benign Cost, C_{ben}, but exploratory messages reduce the likelihood of being attacked. The decreased likelihood of attack is due to the increasing cost incurred by the attacker if discovered as a malicious peer, C_{disc}. The requesting peer can either send an exploratory message or a request message. The serving peer can either respond with an attack or with a legitimate response. In all cases, each peer will incur a cost of C_{ben}.

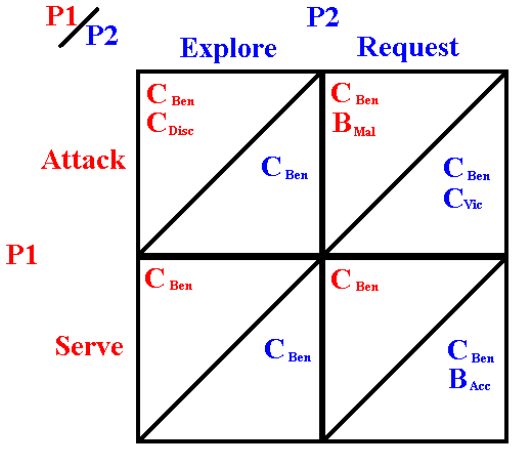

Figure 1. Payoff Matrix for a Benign Peer and a Malicious Peer

Input: Peer Preferences
Output: Resource Access Results
while *Resource Not Accessed* **do**
  Calculate P_{exp};
  Generate Request (P_{exp} are exploratory);
  Send Request;
  if *Request Is Exploratory* **then**
    if *Attacked* **then**
      Blacklist Attacker;
    end
  end
end

Algorithm 1: Algorithm for Sending Exploratory Requests

At this point, we must state some assumptions about the performing resource exploration. First, we have to assume that the peer responding to the request cannot differentiate between exploratory and regular requests. We justify this assumption by noting that a peer can reuse previously obtained results, self-generated results, or pre-programmed results depending on the specific application. Second, we have to assume that the peer sending the requests can verify whether or not an attack has occurred. This assumption is also made in other P2P reputation research [2, 6, 13], since to provide opinions, peers must know whether they were attacked. In many cases, this is manually determined by human users, but due to the cost of human intervention, our work is most useful if the attacks can be automatically determined (for instance, comparing the known checksum of a file to a generated checksum of the file sent by another peer).

Before a peer can decide to utilize the resource exploration, it needs to determine at what rate to send out exploratory messages. In our Adaptive Resource Exploration framework presented in Section 3.4 we show how a peer can improve its utility by learning to play a Nash equilibrium strategy.

2.1 Nash Equilibrium

To determine a Nash equilibrium, the parameters of the game are as shown in Figure 1. By setting the expected value of each action a peer could take equal to the alternative action, the mixed-strategy equilibrium can be determined. As a result, the requesting peer should use exploratory messages with a probability defined by Equation 3 and the serving peer should attack with a probability defined by Equation 4.

$$P_{exp} = \frac{B_{mal}}{C_{disc} + B_{mal}} \tag{3}$$

$$P_{att} = \frac{B_{ben}}{C_{vic} + B_{ben}} \tag{4}$$

An obvious downside to this approach is that it requires a knowledge of the opponent's preferences. AREX overcomes this problem as described in Section 3.2.

2.2 Utility Bounds

The utility-bounded approach to selecting a value for P_{exp} involves two equations. Equation 5 describes the benign peer's average utility per interaction, $Avg_{U_{ben}}$, and

Equation 6 describes the attacker's average utility per interaction, $Avg_{U_{mal}}$.

$$Avg_{U_{ben}} = B_{acc} - \frac{C_{ben}}{(1 - P_{exp}) \times P_{att}} (1 - P_{exp}) \times P_{att} \times C_{vic} \quad (5)$$

$$Avg_{U_{mal}} = (1 - P_{exp}) \times P_{att} \times B_{mal} - \frac{C_{mal}}{(1 - P_{exp}) \times P_{att}} - P_{exp} \times P_{att} \times C_{disc} - C_{ben} \quad (6)$$

To maximize utility, the peer takes the derivative of Equation 5 with respect to P_{exp}, sets the equation equal to 0 and uses the P_{exp} value that produces the maximum utility point (since Equation 5 is quadratic in terms of P_{exp} there is only one maximum).

Equation 5 allows the benign peer to calculate bounds for how high of a exploratory rate it can withstand for the utility it intends to achieve. Furthermore, Equation 6 also allows the peer to predict how much its exploratory rate will reduce the utility of the attacker for a given set of attacker preferences and attack rate.

3. Adaptive Resource Exploration

In this section we present the three main contributions of AREX. These contributions are accomplished by actively adapting to the system the AREX peer is participating in. As a direct result of these contributions, AREX performs effectively in benign, faulty, and hostile environments.

3.1 Faulty Benign Peers

Since Algorithm 1 has no tolerance for inadvertent errors by benign peers, it can reach a deadlock state in which all peers are blacklisted. To overcome this limitation, we present an enhanced version of the algorithm in this section. Rather than blacklisting a peer after any action perceived as an attack, Algorithm 2 is reactive but forgiving. Rather than strictly requiring that a peer can determine if an attack has occurred in all cases, an indeterminable case is permitted. The indeterminable case permits the user to exert an alternate preference of using a peer in the future.

Algorithm 2 allows the peer to define how tolerant it is to attack through the punishment factor, α or the amount of credibility it gives to valid resources through the preference factor, β. Both of these values are non-negative. If the result of an access is indeterminable, then a tolerance factor, χ for the uncertainty is used to evaluate the experience. This value can be zero, positive or negative depending on the disposition of the peer.

The following terms are used in Algorithm 2:

Input: Peer Preferences, Known Peers Experience Vector
Output: Resource Access Results
while *Resource Not Accessed* **do**
  Select Peer i from K with probability $\frac{k_i+1}{\sum_{j=0}^{|K|} k_j}$;
  Calculate P_{exp};
  Generate Request (P_{exp} are exploratory);
  Send Request;
  if *Attacked* **then**
    $k_i -= \alpha$;
  end
  if *Success* **then**
    $k_i += \beta$;
  end
  if *Indeterminable* **then**
    $k_i += \chi$;
  end
end

Algorithm 2: Experience Values

- K: vector containing experience values for the set of available peers
- k_i: experience value for peer i
- α: punishment factor
- β: preference factor
- χ: tolerance factor

3.2 Achieving Nash Equilibrium

The second improvement accomplished by our work is a methodology for playing a Nash equilibrium strategy when insufficient information is available to calculate the mixed-strategy Nash equilibrium described in Equation 3. As a result we have devised an adaptive method for approximating the optimal strategy.

The peer starts by calculating the opponent's Nash equilibrium point for P_{att} by using Equation 4. Then the peer uses Equation 5 to calculate the P_{exp} that will result in the highest initial expected utility; however, a peer cannot maintain this strategy, because a strategic peer will adapt its strategy to exploit naivety. Additionally, if the opponent is not playing a Nash equilibrium, and is instead playing a sub-optimal strategy (or is benign), the peer wants to exploit that information to its advantage.

To adapt to the environment, the peer continually adjusts its estimation, P'_{att} of what P_{att} is by using equation 7 where ϕ is a discount value (to prevent overreaction) and γ is the indeterminable discount in the range of $0 \leq \gamma \leq 1 - \phi$. Based on the new value of P'_{att}, the strategy is recalculated.

$$P'_{att} = \begin{cases} \phi \times P'_{att} + (1 - \phi) & \text{if attacked} \\ \phi \times P'_{att} & \text{if not attacked} \\ \phi \times P'_{att} + \gamma & \text{if indeterminable} \end{cases} \quad (7)$$

In Section 4.3 we discuss the effectiveness of this approach in approximating a Nash equilibrium and strategies for sub- optimal attackers.

3.3 Strategy Selection

We describe our third improvement: Handling differences in strategy between individual peers and the system as a whole. In our preliminary work, each peer can be treated individually or the system as a whole when computing P_{exp}, but there was no methodology for deciding which approach was appropriate.

strategic peers and a heterogeneous population of peers. If all peers were to attack (or be unreliable) at a fixed rate then the best strategy would be to to estimate that attack rate and maximize utility by always sending real requests if the attack rate was less than $\frac{B}{C}$; however, since rational, strategic peers seek to optimize their own utility, they will adapt to our strategies which could then cause our strategy to become ineffective.

AREX may operate in systems of heterogeneous attackers. Some may adapt to the AREX strategy while others attack at a fixed rate. This situation begs the question: How can strategic attackers be motivated to attack less while simultaneously utilizing benign and non-strategic attackers?

We argue that the question is actually addressed by the two previous solutions. We assume the worst-case scenario: The entire system is colluding against the AREX peer. In this case, we can treat the entire system as a single attacker. If a peer is not in the worst case scenario, but instead, sub-optimal attackers and benign peers exist in the system, then the two-level approach described in Section 3.1 allows the AREX peer to decrease the probability of being exploited by strategic attackers and sub-optimal attackers with a propensity for attacking. The approach results in the system being transformed in such a way that a peer can still treat the system of other peers as a single attacker whose attack strategy is described by Equation 8 where P^i_{att} is the probability that peer i will attack.

$$P_{att} = \sum_{i=0}^{|K|} \frac{k_i}{\sum_{j=0}^{|K|} k_j} \times P^i_{att} \quad (8)$$

Consider the following three cases:

- The system is mostly benign
- The system consists of sub-optimal attackers
- The system consists of optimal attackers

In the first case, the probability of requesting from any sub-optimal attackers would approach 0, as would the probability of requesting from optimal attackers. Then the system would be modeled by an attacker similar to that of the benign peer. In the second case, the exploratory messages would identify the sub-optimal attackers with the least propensity for attacking. Then the attack strategy of the perceived system would tend toward the the malicious peers with the lowest attack rates when benign peers were not available. In the third case, the system would model the optimal attackers as a single attacker. If heterogeneity was introduced in terms of types of peers in the last two cases, Algorithm 2 would tend toward accessing resources from peers with the lowest attacking rates. Finally, in the case that no peer was distinguishable from another (or no peer was ever accessed multiple times), our approach will treat the entire system as the average of its members.

3.4 Example

Figure 2 shows AREX in operation. In this example we view the effects of Peer1's actions of attacking, serving, and an indeterminable response. These actions occur in a system where the AREX peer is connected to three other peers. At the beginning of this scenario, all three peers have K_i = 5 (5 valid responses) and P^i_{att} = 0 (no attacks). When Peer1 attacks the AREX peer, Peer1's K_i value decreases by α, decreasing its probability of being used for the next access from $\frac{1}{3}$ to $\frac{1}{11}$. Upon the successful serve of the next request, Peer1's K_i value increases and its P^i_{att} value decreases; however, the overall P_{att} estimate for the system increases because of the increased chance of selecting Peer1. Finally, after the indeterminable result, the probability of selecting Peer1 decreases, and its P^i_{att} value remains the same, thus reducing the system's estimated P_{att}.

4. Simulation Setup and Results

We now describe simulations designed to test the ability of AREX peers to improve their utility and reduce attacks against them. Our simulations model both static and mobile systems and test utility and energy costs against several attack models. They show that AREX provides substantial benefits to users and creates incentives for attackers to limit their malicious behavior.

4.1 Simulation Setup

We developed a discrete, time-stepped, simulator at the level of resource accesses. At each step, the AREX peer executes Algorithm 2 and sends a request. After receiving the

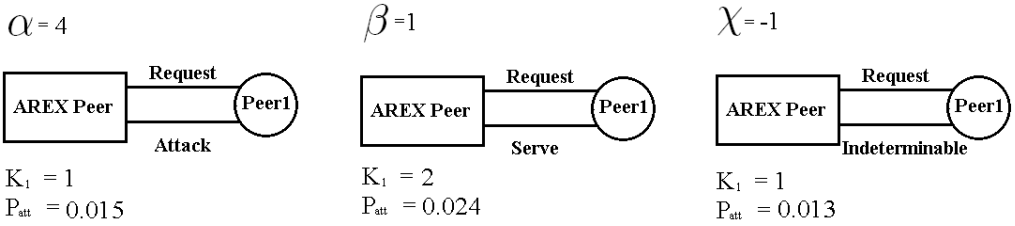

Figure 2. AREX Example Behavior

Number of Peers	1000
Mobility	0%
Connectivity	100%
Avg Benign Reliability	95%
Execution Time (seconds)	1000
Attack Rate of Malicious Peers	100%
C_{ben}	1
C_{vic}	100
C_{mal}	0
C_{disc}	1
B_{acc} (static peers)	120
B_{acc} (mobile peers)	150
B_{mal}	100
α	1
β	1
χ	1
ϕ	.95
γ	0

Table 2. Default simulation parameters

result, the peer recalculates the values k_i and P'_{att}. If mobility is enabled, then mobile peers are modeled as moving randomly.

All attackers have the same preferences, as listed in the chart above. Benign peers always try to return the proper response, but may fail or lose connectivity during service. Note that peers who are unwilling to provide service, i.e. *freeriders*, are a different problem and beyond the scope of this work. Thus we model the reliability of benign peers as normally distributed with a mean of 95% and a standard deviation of 1%.

4.1.1 System Model

We first simulate a decentralized and unstructured Peer-to-Peer (P2P) system. This system is static, meaning that the peers are completely immobile, and stable, meaning there is no node churn. The number of peers is set at 1000.

We also simulate a mobile system, presented in Section 4.3.2. All parameters are the same as before, except peers enter and exit the range of the AREX peer at a variety of rates. In these simulations, we vary the peer mobility of both moving in and out of range at rates varying between 1% to 100%. The peers operate within a frame of reference relative to the AREX peer. A rate of 1% represents a peer that is unlikely to change its current position relative to the AREX peer and a rate of 100% represents a peer that will always change its current position relative to the AREX peer at each time step. The peer direction traveled is randomly selected from a uniform distribution, and the distance traveled is always a distance great enough to cause disconnectivity from the AREX peer if the moving peer was currently connected to the AREX peer. The choice to model the mobility of other peers randomly instead of with a travel pattern was made in order to simulate a situation in which it would be more challenging for the AREX peer to adapt.

4.1.2 User Model

We use two user models for comparison. Our first model depicts a naive user who always attempts to access resources. The second model applies the AREX mechanism to access resources.

4.2 Attacker Models

The attacker model defines the benefits and costs to the attacker. In our attacker model, the system attacks some percentage of the time ($f\%$). In addition to f being an arbitrary percentage, we also present results for when f is a special percentage in the system representing the following.

- Always attack
- Attack with a Nash equilibrium
- Attack at variable rates
- Never attack

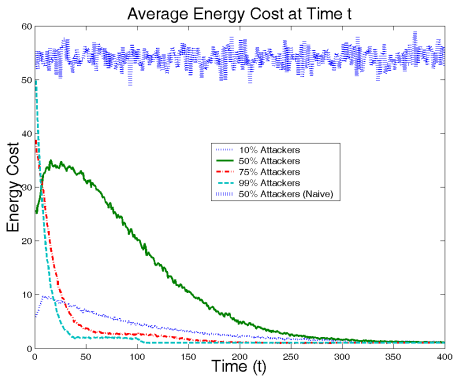

Figure 3. Effect of AREX Adaptation Against Various Attack Rates

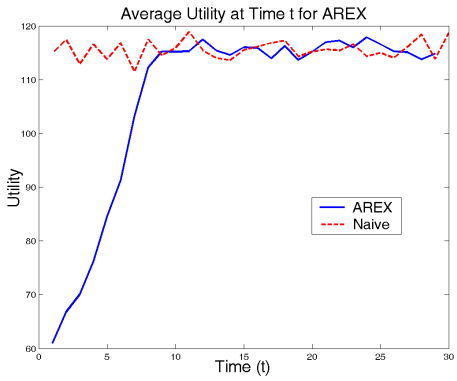

Figure 4. AREX adapting to a Mostly Benign System

Attackers are assumed to be consistent in terms of preferences. For example, if an attacker has a B_{mal} of 50, then its B_{mal} will be 50 throughout the entire time of the simulation. The attackers used, as noted by the table above, are based on a powerful and motivated attacker in order to show that our mechanism works against a strong opponent. If we were to model an attacker with less reward for being malicious and more cost for being malicious, then the performance of the mechanism would further improve.

4.3 Results

In the remainder of this section we present and discuss the simulation results of AREX-based resource access. We present results over the simulation time that demonstrate AREX's ability to perform with increasing effectiveness as i) parameters vary and ii) user preferences vary. Unless otherwise noted, the simulation parameters are as shown in Section 4.1. In all cases, each simulation was run 1000 times and the average of those runs is presented.

4.3.1 Time-Based Results

First, we examine the average energy cost at a given time during the execution of our system in Figure 3. In this figure, each point on the plot represents the average cost at the respective timestep. For reference, the average cost of a naive approach when 50% of the system is attack is also given. As the figure shows, the average energy cost at a given point in time and approaches the benign cost of participating in the system, even when 99% of the system is attacking. This means that AREX adapts to attackers and learns to decrease the expected cost as the system persists. The reason for this is that as AREX determines who the attackers are, they receive less opportunity to attack. In the 99% case, these attackers are quickly identified and receive only a minimal number of requests, leaving the bulk of the requests to be directed toward reliable, benign peers. Hence, AREX results in a low cost steady-state system.

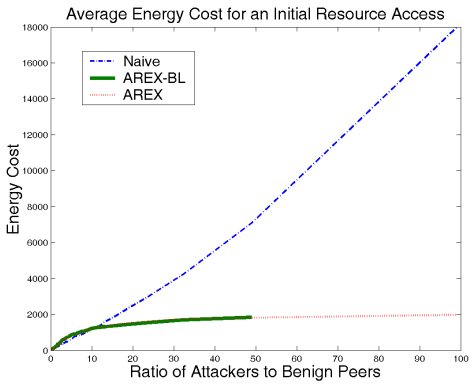

Figure 5. Average Cumulative Cost to Access First Resource

4.3.2 Resiliency Results

We discuss the performance of AREX as we vary the percentage of attackers (hostility), the number and set of peers known at any given time (mobility), and the number of peers in the system. Mobility was simulated by varying the rate of mobility per time step. At each time step, each peer randomly moves relative to the AREX peer at the rates shown in Figure 6.

Figure 5 shows the average cumulative energy consumed to access a first resource. The x-axis shows the ratio of malicious peers to benign peers in the system. If a peer is not

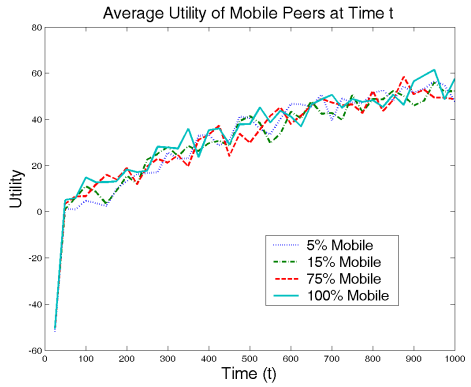

Figure 6. Average Utility Over Time for Mobile Peers

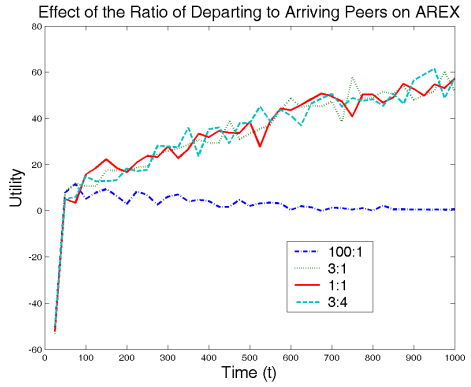

Figure 7. Effect of Arrival and Departure on Average Utility Over Time for Mobile Peers

malicious then it is unreliable on average, 5% of the time. The plot of AREX-BL (The Blacklisting version of AREX) stops before the other two plots because in some simulations the peer fails to access a resource at all in the unplotted situations, thus demonstrating the need for the tolerant version of AREX. The overhead associated with resource exploration only increases the energy costs a small amount over a naive access strategy when the system is mostly benign, and when the system becomes overwhelmingly malicious, the energy savings of AREX become immense.

In the simulations used to obtain the data in Figure 6 and Figure 7, the location of each mobile peer was updated at each time step. The results of the mobility simulations reveal that AREX is not negatively affected by mobility, as in Figure 6. This simulation occurred with half the system as attackers and an equal distribution of attackers and benign

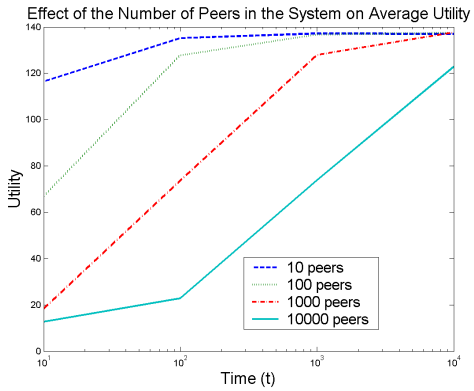

Figure 8. Effect of Number of Peers on AREX Performance

peers being mobile. The more interesting factor was the ratio of peers leaving the range of the AREX peer to peers entering the range of the AREX peer as shown in Figure 7. This figure shows AREX's performance is not diminished for similar arrival and departure rates; however, when the number of peers returning to the range of the AREX peer is insignificant compared to the number leaving, utility is diminished, but it must be an extreme case as demonstrated by the $100:1$ ratio. The reason for the diminished utility is that AREX has less benign peers to adapt to using more frequently at any given time.

Next we present our results showing AREX's resilience to attack as the number of peers in the system increases. Intuitively, as the number of peers increases, the longer it takes AREX to adapt. This intuition is shown to be true in Figure 8 which shows the steady state utility per request of systems with varying number of peers. The cause of the steady state is the AREX peer's adaptation to the system identifies a small group of peers that provide good resources. Because AREX does not rely solely on that group, but instead randomly selects peers outside of the group (though less often than in the group) steady state utility takes longer to converge when there are more peers outside the group to explore. By varying the β parameter, the AREX peer could force more preference for the early members of the group and cause a quicker convergence to steady state utility.

4.3.3 Preferential Results

The results discussed here assist us in determining what situations it is appropriate to use AREX and to what extent it will be effective. We have identified the ratio of B_{acc} to C_{vic} (AV ratio) and the ratio of B_{ben} to C_{vic} (BV ratio) as the two important preference factors for AREX. We

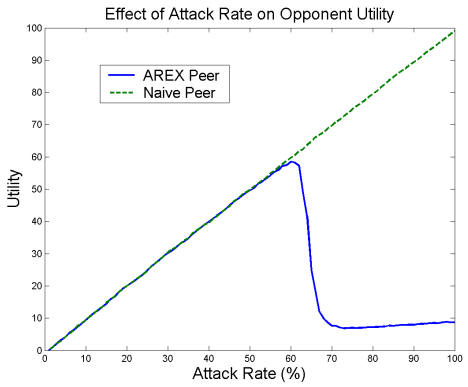

Figure 9. Effect of AREX on Opponent Preference to Attack

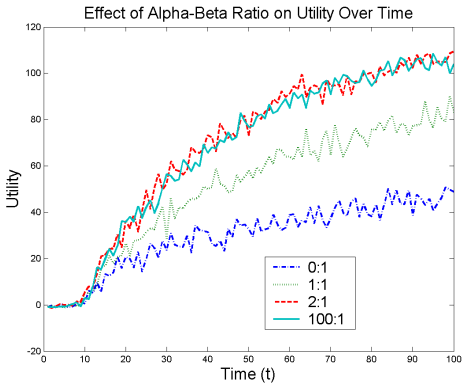

Figure 10. Effect of α:β Ratio on AREX

also show the use of AREX to decrease a strategic attacker's preference for attacking.

Figure 9 shows the effect of AREX on an attacker's utility. The graph shows that while AREX has an insignificant effect on the attacker when the attacker plays a strategy when it attacks less than the Nash equilibrium strategy. When the attacker plays a strategy greater than the Nash equilibrium, its utility is greatly diminished. Hence the attacker is motivated to attack significantly less as a result of the AREX. The analytically computed Nash equilibrium point for the attacker is approximately 0.643 and the simulation results show the optimal utility to be within a few percentage points of this value. The values differ as a result of the benign peer adjusting to learn its optimal strategy since it does not know the attacker's strategy *a priori*. This allows the attacker to achieve a maximum utility with a rate that is slightly different from the Nash equilibrium.

The results shown in Figure 5 vary as we change preferences as noted in Section 4.3.2. The change in results occurs based on the ratio of C_{ben} to C_{vic} (RV ratio). As the RV ratio increases, the intersection point (the point that defines when it is in the peer's best interest to change the rate of exploratory messages) also increases with respect to the percentage of the system that is malicious. The RV ratio also affects the difference in costs between any two exploration rates. As the RV ratio approaches 1, the difference in cost between any two strategies as the percent of the system is attacking changes approaches 0. As the RV ratio approaches either zero or infinity the cost difference between any two exploration strategies approaches infinity.

Figure 10 shows that the ratio of α to β has little effect on the performance of AREX pending that the ratio is greater than 1. When the α:β ratio was greater than one, the simulations converged to the same average utility; however, when the ratio was 1 (meaning punishment and reward are the same) or 0 (meaning there is no punishment, only reward), the utility derived by the AREX peer was greatly diminished.

5. Related Work

Trust, in the form of reputation management[9, 8, 6, 10], and incentives[5] have largely been a focus in P2P security from an end to end perspective. Reputation systems focus on accumulating reputations and propagating them through the network, so other peers can interpret the reputations to make decisions on who they should trust. Incentive solutions provide some form of payment to peers to encourage good behavior. A problem with reputation systems is that they require prior knowledge to work. In other words, peers are vulnerable to attack if they do not have knowledge or correct knowledge of other peers in a reputation system. As stated in the introduction, the vulnerability is most evident when a peer first enters a system or a peer previously recognized as benign chooses to betray trust (or is compromised). Since that peer would have a good reputation up until that point, a reputation system would give no reason not to trust that peer. Incentive systems are vulnerable because they do not prevent attack, they just give more reason to cooperate in the system, but the vulnerability is still there if the malicious peer prefers acting maliciously enough.

Research in economics, particularly utility functions and game theory, has had a large influence in computer science. While much of the research is focused on auctions, some similar concepts that are discussed in this paper are being researched [1]. In particular, economic-based approaches have permeated both security [5, 11] and P2P computing [7, 4, 5]. These solutions do very little to address general malicious behavior in P2P systems. Instead, those related

to P2P systems are largely focused on incentives to prevent freeloading.

6. Conclusion

In this paper we present a novel adaptive mechanism called AREX for secure resource access in P2P environments. In AREX, benign peers send exploratory messages to assess the actions of unknown peers and quickly adapt based on the reactions of peers. AREX adaptively balances the trade-off between exploration and utilization of resources to protect the peer running it with minimal energy costs. AREX is found to perform well especially in dynamic environments where previous work was found to be inadequate. AREX has been shown to perform well in both hostile and benign environments. Furthermore, AREX not only benefits the peer running it, but it also reduces rational attacker's motivation to attack by playing an approximate Nash equilibrium strategy against the attacker. As a result, AREX provides the most improvement over non-AREX peers in hostile environments. Simulation studies validate our findings and demonstrate the superior performance of AREX in terms protecting benign peers, rendering malicious peers ineffective and energy costs.

In the future, we will extend AREX to challenged environments, including sensor systems, opportunistic networks. Utilizing AREX capabilities we are in the process of developing a framework for distributed trust in dynamic environments.

References

[1] N. Nisan and A. Ronen. Algorithmic Mechanism Design. In The Thirty First Annual ACM symposium on Theory of Computing, pages 129-140, May 1999.

[2] S. Kamvar, M. Schlosser, H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[3] L. Chen and P. Pu. Survey of Preference Elicitation Methods. A Technical Report. 2004.

[4] J. Chuang. Economics of Peer to Peer Systems. Academia Sinica 2004 Summer Institute on P2P Computing, 2004.

[5] M. Feldman, K. Lai, I. Stoica, J. Chuang. Robust Incentive Techniques for Peer-to-Peer Networks. In the Proceedings of EC, 2004.

[6] L. Xiong, L. Liu. PeerTrust: Supporting Reputation Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering, Vol 16, No. 7, 2004.

[7] D. Figueirdo, J. Shapiro, D. Towsley. Incentives to Promote Availability in Peer-to-Peer Anonymity Systems. Proceedings of the 13th IEEE International Conference on Network Protocols, 2005.

[8] M. Srivatsa, L. Xiong, L. Liu. TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks. In Proceedings of WWW, 2005.

[9] S. Song, K. Hwang, R. Zhou, Y. Kwok. Trusted P2P Transactions with Fuzzy Reputation Aggregation. IEEE Internet Computing Magazine, Nov/Dec 2005.

[10] G. Suryanarayana, J. Erenkrantz, and R. Taylor. An Architectural Approach to Decentralized Trust Management. IEEE Internet Computing Magazine, Nov/Dec 2005.

[11] R. Anderson and T. Moore. The Economics of Information Security. Science 314, pp 610-613, October 27, 2006.

[12] S. Marti and H. Garcia-Molina. Taxonomy of Trust: Categorizing P2P Reputation Systems. Computer Networks: The International Journal of Computer and Telecommunication Networking, vol 50, pp. 472-474, March 2006.

[13] K. Walsh, E. Sirer. Experience With A Distributed Object Reputation System for Peer-to-Peer Filesharing. In Proceedings of the Symposium on Networked System Design and Implementation (NSDI), San Jose, California, May 2006.

[14] K. Hoffman, D. Zage, and C. Nita-Rotaru. A Survey of Attack and Defense Techniques for Reputation Systems. Purdue CSD TR #07-013, 2007.

[15] A. Josang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, vol 43, pp 618-644, March 2007.

[16] B. Lagesse and M. Kumar. A Novel Utility and Game-Theoretic Based Security Mechanism for Mobile P2P Systems. IEEE Pervasive Computing Workshops, Mobile P2P, 2008.