Nguyen Tran – Research Statement

My research interests are in networked systems with a focus on security and privacy. Our society is entering an era where many aspects of computing are being outsourced to different entities. For example, companies and users outsource the tasks of data storage and computation to cloud platforms and third-party application developers. Popular websites such as YouTube and Reddit outsource content and opinion generation to individual users. While such outsourcing drastically decreases the cost of building and running a popular service, it also introduces significant security and privacy risks: if applications or cloud platforms become untrustworthy, users can have their sensitive data leaked without permission. If attackers generate and aggressively promote bogus material, cooperative websites will become inundated with spam and malicious content. The goal of my research is to develop security mechanisms for outsourced systems which enable enforcement of privacy policies and mitigate the damage caused by attacks.

To harden traditional systems against attacks, research has been done on constructing software according to the least-privilege principle, eliminating security-related software bugs, and performing robust authentication and access control. Securing outsourced systems calls for a fundamentally different approach. None of these traditional approaches is effective, as outsourced systems are dependent on user-generated inputs or untrustworthy physical platforms outside of an entities control. My research aims to secure outsourced systems by helping them assess the reputation of user-generated inputs and attest the security-compliance of remote platforms. In the rest of this statement, I describe my thesis research on securing web communities and peer-to-peer systems, and discuss my recent work on building a privacy platform for cloud computing.

1 Thesis research: security and incentive for cooperative systems

Nowadays, many popular systems are outsourcing content generation and propagation to users. For example, users submit and comment on various types of content such as news articles (Digg), Q&A (Quora), pictures (Flickr) and videos (YouTube). In peer-to-peer content distribution networks such as BitTorrent, users contribute bandwidth resource to help each other download files. These systems can benefit immensely from this outsourcing approach as long as users cooperate. However, since adversaries can join the system easily and many users are selfish, such *cooperative systems* are plagued with various security and incentive problems. For example, adversaries have manipulated the voting system of Digg to promote their articles of dubious quality [7]. Selfish users in public BitTorrent communities leave the system to avoid uploading files to others, resulting in drastic performance degradation for these content distribution systems [5]. A robust cooperative system must be able to limit the amount of damage inflicted by adversaries and to incentivize honest users to make adequate contributions.

My Ph.D thesis research aims to create practical security and incentive primitives for cooperative systems. The main theme of my work is to leverage the social network among users in designing secure and incentive-compatible cooperative systems. Existing cooperative systems lack strong user identities, and suffer from Sybil attack and white-washing attacks as a result. In a Sybil attack, an adversary creates many fake identities and uses them to amplify his attacking power. In a white-washing attack, an adversary replaces a misbehaving identity with a new identity to avoid any penalization of his misbehavior. Robust identity management is the key to repelling both types of attacks. Existing systems restrict the creation of new identities based on IP addresses or the ability to solve CAPTCHA puzzles. However, these approaches are not effective as an adversary can easily obtain different IP addresses using public proxies and the solving of CAPTCHA's can be automated at very low cost. My work builds upon the insight that real world social relationships require significant effort to foster social relationships. Therefore, I can treat social links as a scarce resource and use them to manage user identities in ways that are resilient to Sybil and white-washing attacks. Next, I will describe how I exploit this insight in building various systems that address the security and incentive problems in online content voting, admission control, and peer-to-peer content distribution.

Sybil-resilient online content voting [7]: Many cooperative websites use votes from users to rank user-generated online content. However, in most content voting systems, an adversary can easily out-vote real users by creating many Sybil identities. I designed an online voting system, named SumUp, that leverages the social network among users to defend against the Sybil attack. Because social links are a scarce resource, the adversary is unlikely to possess many attack links with honest users. However, the adversary may create many links among Sybil accounts.

To limit the voting power of the adversary, SumUp collects votes from users by computing a set of max-flow paths on the social graph from all voters to a set of trusted identities. In order to perform max-flow computation, one needs to set the flow capacity of each social link. One possible assignment is to set the capacity of every link to be one. Under such an assignment, the max-flow based vote collection can bound the attack capacity between the set of trusted vote collectors and the adversary to be the number of attack links (k), thus collecting at most k bogus votes from the adversary. However, this assignment also prevents most honest voters from having their votes collected, as the flow capacity between honest voters and trusted vote collectors is limited by the small number of immediate neighbors of the vote collectors.

I devised an assignment strategy which gives links close to vote collectors relatively higher capacities than links that are far away. Such an assignment enables max-flow based vote collection to gather most honest votes while still limiting the number of bogus votes to be the number of attack links with high probability. SumUp's defense for this vote aggregation problem is asymptotically and an order of magnitude better than other Sybil defense techniques [8]. SumUp also incorporates users' feedback on the validity of collected votes to further improve its performance. In particular, if the adversary is found to have casted many bogus votes, SumUp eventually eliminates the adversary from the social network. SumUp offers immediate benefits to many popular websites that currently rely on users' votes to rank content. I applied SumUp on the social graph and voting trace of Digg and found strong evidence of Sybil attacks. In particular, I identified hundreds of suspicious articles that have been promoted to the "popular" status on Digg by possible Sybil attacks.

Sybil-resilient admission control [6]: SumUp is a centralized system designed to collect the votes of users after they have voted on some piece of content. Another approach in building a Sybil-resilient voting system is to admit new identities into the system in a Sybil-resilient fashion and only allow admitted identities to vote. In fact, besides online voting, most cooperative systems including online user communities and peer-to-peer applications also benefit from restricting the number of admitted Sybil identities. The wide applicability of Sybil-resilient node admission control has motivated me to design GateKeeper, a distributed protocol which performs node admission control based on the social graph among users. GateKeeper admits most honest nodes into the system while only allowing a small constant number of Sybil identities per attack edge. GateKeeper improves over previous social-graph based admission control [Sybillimit Oakland'08] by a factor of $\log n$, where n is the number honest nodes in the social graph, and in fact, can be proven to be optimal in the number of Sybil identities admitted.

Collusion-resilient reputations for P2P content distribution networks [5]: Apart from containing the damage of malicious activities, cooperative systems must also incentivize users to contribute. Among today's cooperative systems, peer-to-peer content distribution networks can especially benefit from an incentive mechanism to encourage nodes to contribute their upload bandwidth. BitTorrent provides incentives for nodes to upload to each other if they are currently downloading the same file. However, when nodes finish downloading, they no longer have incentives to upload to others. As a result, the practical performance of BitTorrent is dramatically less than its achievable potential. I performed measurements that show that average download bandwidth in private BitTorrent communities is $8 - 10$ times higher than that in public BitTorrent communities, primarily as a result private BitTorrent communities having more incentivized seeders [5]. However, the ad hoc mechanisms that are used in private BitTorrent to incentivize seeding are vulnerable to attacks.

To address this problem, I have designed the Credo reputation system which encourages upload contribution by providing higher download speeds to nodes with higher reputation scores. I implemented Credo in the Azureus BitTorrent client. Credo reputation correctly captures a user's past contribution and is resilient to Sybil attack and collusion. In Credo, nodes give credits to uploaders when downloading data from them, and collect credits by uploading to others. Credo reputation score is computed from the credits that the node has collected from others. The reputation is computing using two techniques: counting the diversity of the credits, and modeling good behavior. These techniques ensure that the maximum reputation score of an adversary is bounded by a constant which is independent from the number of adversaries in the colluding group. Furthermore, the reputation scores will decrease as the adversaries keep downloading more files.

2 Security and privacy for cloud computing

In many cooperative systems, incentive is not a concern if one can outsource physical resources to the cloud. For example, I leveraged real-world relationships in order to incentivize users to contribute storage and upload capacity for a cooperative P2P backup and file-sharing system called Friendstore [2]. A system designer does not need to address such incentive concerns and can focus on other aspects of the system if the system provider or users can afford cloud service like the case of DropBox. Cloud computing has provided system designers with significant new functionalities and has spurred innovation. However, security and privacy are still big concerns which discourage system providers and users from moving their computing infrastructure to the cloud. A recent survey by Microsoft revealed that "...58% of the public and 86% of business leaders are excited about the possibilities of cloud computing. But, more than 90% of them are worried about security, availability, and privacy of their data as it rests in the cloud." Today, when a data owner outsources her data into the cloud, she effectively loses control over it. Regaining control over users' data in the cloud is essential for cloud computing to be adopted widely in the future.

There are several approaches in addressing this problem. While "encryption" is the magic catch-all in the security and privacy realm, data encryption alone does not address this problem completely because users need to perform rich computation on their private data. Although recent advance in cryptography allows arbitrary computation on encrypted data, such techniques, i.e. fully homomorphic encryption, are still far from practical. Moreover, existing applications need to change significantly in order to adopt homomorphic encryption; debugging a computation performed on cipher text is also difficult. I believe a viable solution should incur reasonable system overhead and provide ease of development and deployment. In an effort to find such solution, I spent the last six months at UC Berkeley collaborating with Prof. Dawn Song and her research group on designing and building a Platform for Private Data (PPD) for cloud computing [1]. My previous experience in building backend infrastructure, e.g. online migration for geo-distributed storage systems [4] and auto-tuning buffer management for database [3], has benefited the design of PPD.

PPD is a software stack meant for various cloud providers to run on their own hardware infrastructure. In PPD, users data objects are encrypted, and annotated with their respective access control list (ACL), forming secure data capsules. An untrusted application that runs on PPD consists of many modules which are categorized as two types: *application front-end* and *storage backend*. These untrusted modules access decrypted data (made available by PPD) and perform rich computations on them only inside isolated containers. Any derived data resulting from such computation is encrypted again under an appropriate key by PPD upon exiting a container. Storage backend modules can access private data from many users in order to provide storage optimization and service such as deduplication, compression, and replication. However, application front-end modules can only request data from the storage backends through *trusted storage proxies* which are provided by PPD. Since the storage proxies provides simple interfaces such as key-value-store and file system, they can check the integrity of the data capsules and the associated ACLs returned by the untrusted storage backends, thereby returning the capsules to a front-end module only if it has an appropriate security context. The security context of each front-end module is updated as it accesses more data capsules. The PPD strictly enforces communication between untrusted front-end modules in order to preserve users' access control policies using information flow control. Finally, we rely on the trusted hardware TPM to attest that the trusted PPD has been executed correctly on the machines in the cloud.

PPD enables a new cloud computing paradigm that bridges the gap in trustworthiness between three principles: end users, application developers, and cloud providers. First, end users can be assured about the security and privacy of their data, meanwhile benefiting from a richer spectrum of applications from diverse sources that they might hesitate to use otherwise. Second, developers can focus on innovation and agile development, without having to be security experts, and still offer user data protection for their applications. Finally, cloud providers can prove to other principles that PPD is correctly run on their physical machines through code attestation technology.

3 Current projects

I believe that the work that I have done is only an early contribution in the research area of security and privacy for outsourced systems. The importance of cooperative systems will continue to grow in the future. There are increasingly more attractive domains for adversaries to exploit. PPD is just a first step in regaining full user control over their data in the cloud. There are still many privacy and systems aspects that need to be addressed. I expect to push the limit of existing designs and to contribute new techniques in this area. Below are my ongoing research projects.

Combating information censorship: Recent events during the so-called ”Arab Spring” have shown the power of the Internet when it comes to organizing large groups of people for protests. Such events have also shown how willing repressive governments are to censor the Internet or to disconnect their populace from the Internet entirely such as the incidents in Egypt and Libya in early 2011. I believe that during these government-imposed communication blackout periods, people should still be able to exchange information and organize among themselves. My goal is to design and build a networked system to help the citizens in such countries to disseminate information during such blackouts in the presence of an adversarial government.

One promising solution is that the citizens in these countries form a cooperative system in which citizens use their mobile devices to exchange messages with nearby devices through WiFi during communication blackout. This opportunistic communication is promising since it is hard for the adversarial government to block it. However, the adversarial government can still create Sybil identities and use them to flood the system with bogus (junk) messages, or confuse the citizens by generating dishonest votes on authentic messages. The defense provided by SumUp can potentially prevent this threat. However, SumUp requires each citizen to know the entire social graph in order to filter out bogus messages. This raises privacy concern since the government can perform deanonymization to ascertain the identity of the protesters, thus to persecute them. I am designing a enhanced version of SumUp so that it does not require the knowledge of the entire social graph, thereby addressing this privacy concern.

Reputation-based routing: Secure routing protocols that do not rely on a public-key infrastructure are desirable because of the relative ease of deployment. For example, deploying a PKI for inter-domain routing over the Internet is a serious challenge. In the BGP setting, we can view the network of autonomous systems (AS) as a cooperative system in which the ASes are sharing the view of the network through exchanging routing tables. The relationships among ASes can also be viewed as a trust graph where the adversary has only a few links with other ASes. However, the adversary can generate and propagate a large amount of bogus routing information in order to hijack traffic. My goal is to minimize the number of incorrectly computed routes by honest ASes due to bogus routing information. A promising solution that I am investigating right now is to embed a decentralized reputation mechanism into existing routing protocols such as path-vector and link state protocols to reduce the risk of choosing routes via the adversary.

Toward Data Protection as a Service for cloud computing: PPD is only the first step for cloud computing to fully provide Data Protection as a Service (DPS) in addition to providing hardware and software platforms. DPS is not only desirable for both end users and application developers as mentioned previously; it can potentially enable a new ecosystem in which end users can have full control over how to share their private data in exchange for monetary benefit. For example, a suppose marketing company may want to know the average age of a group of users. A user may want to help the company to compute such statistical information in exchange for monetary compensation, but still does not want to reveal her real age to the company. Such a scenario can be supported if PPD provides a mechanism for differential privacy and allows users to specify more complicated policies on their data capsules. Currently, PPD only supports simple ACL on data capsules. How to enable differential privacy and more complicated privacy policies on PPD? How to support debugging for application developers without revealing users' private data? Users usually share data with their friends. Can we leverage this sharing pattern to place data in the cloud in order to improve access locality for applications? I want to address these questions on privacy and systems aspects of PPD in the near future.

References

[1] Krste Asanovic, Petros Maniatis, Prashanth Mohan, Charalampos Papamanthou, Elaine Shi, Dawn Song, Emil Stefanov, Mohit Tiwari, and **Nguyen Tran**. PPD: A platform for private data. In submission.

[2] **Dinh Nguyen Tran**, Frank Chiang, and Jinyang Li. Friendstore: Cooperative online backup using trusted nodes. In *Proceedings of International Workshop on Social Network Systems (SocialNet)*, 2008. Extended version will appear in ACM Transactions on Storage (TOS).

[3] **Dinh Nguyen Tran**, Phung Chinh Huynh, Y. C. Tay, and Anthony K. H. Tung. A new approach to dynamic self-tuning of database buffers. *Transactions on Storage (TOS)*, 4(1):1–25, 2008.

[4] **Nguyen Tran**, Marcos K. Aguilera, and Mahesh Balakrishnan. Online migration for geo-distributed storage systems. In *Proceedings of USENIX Annual Technical Conference (USENIX ATC)*, 2011.

[5] **Nguyen Tran**, Jinyang Li, and Lakshminarayanan Subramanian. Collusion-resilient credit-based reputations for peer-to-peer content distribution. In *Proceedings of International Workshop on Economics of networked systems (NetEcon)*, 2010.

[6] **Nguyen Tran**, Jinyang Li, Lakshminarayanan Subramanian, and Sherman S.M. Chow. Optimal sybil-resilient node admission control. In *Proceedings of International Conference on Computer Communications (INFOCOM)*, 2011.

[7] **Nguyen Tran**, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *Proceedings of USENIX symposium on Networked systems design and implementation (NSDI)*, 2009.

[8] Haifeng Yu. Sybil defenses via social networks: a tutorial and survey. *SIGACT News*, 42:80–101, October 2011.