Peer-to-Peer Networking

P2P Security, Fairness and Trust

Lecture Content

- Different risk scenarios in P2P environments
- Security requirements concerning peers
  - Authentication, authorisation and data integrity
- Security issues in structured P2P Systems
  - Storage and routing
  - Various security algorithms and schemes used in P2P
- Fairness and trust
  - Reputation and trade mechanisms

Security

- *“Three little pigs have a lot to teach us about security. The successful pig didn't have a core competency in wolf detection but had one in house building. When the wolf is at the door, it's too late to start building a sturdy house.”*

Security in P2P

- Regarding to P2P, the user requirements for security boil down to three different statements
  - I need to be sure that the person I am communicating with is who he or she claims to be
  - I need to control when and how people access resources on my system
  - I need to have confidence that my messages are not read or modified en route

Why Is Security Such a Big Deal for P2P

- Besides the basic risks from merely being connected to the Internet...
- Peers act as servers, and their system offers services
  - Cannot know accessing entities
- Peers offer resources on their systems for use
  - Protecting non-shared parts, manage the level and mode of access
- Average user has no prior experience in managing a server or a service

Potential Risk Scenarios

- Direct communications including general file sharing
  - Peers installing new applications on their host machines
- P2P communication may tunnel through a firewall
  - Passing content cannot be monitored, firewall is useless
- Requesting a download based on blind faith
  - You get what you asked for?
- Non-secure application lets peer to snoop around
  - Steal content, online identify from browser cookies, etc.

Security in P2P Applications

- Varies from nothing to strong authentication and full encryption of disk space and communications
  - Consider most (all?) P2P file sharing networks insecure
  - Commercial P2P based on known protocols, such as SSL and HTTPS
- The minimum security measures should involve
  - Authentication: be certain of the peer's identity
  - Authorisation: control what others can do on your system
  - Data integrity: Data you send or receive is not tampered

Authentication

- “Something that you know”
  - User identifier and password
  - Considered a weak authentication
  - Hard to manage in P2P without centralisation
- Add to previous “something you have”
  - Compare to a card with a PIN number
  - Digital certificate and a private key
  - Known as two-way authentication, most commonly used in P2P security environments

Public Key Security Handshake

- Mutual authentication
  - Both ends of exchange wish to authenticate the other
- Public key security handshake as an example
  - Both parties have digital certificates signed by a trusted authority (CA) and CA's certificate containing public key
  - Alice wants to connect Bob: she sends her certificate (identity, public key and CA)
  - Bob checks that CA's digital signature is valid
  - Now Bob needs to make sure that Alice is verified

Public Key Security Handshake

- Verification is done by sending a random message to Alice for encrypting with her private key
- If Bob can decrypt the message correctly by her public key, Alice can be trusted
- Now the same procedure is done in reverse
  - Alice can identify and verify Bob
- No need to reveal passwords, but need to trust CA

Authorisation

- Process of giving someone permission to do or have something
- Generally, the granting peer has a *access control list* (ACL) containing permission granted to other users
  - Check the request against ACL
- Alternatively permission tracking can be placed within the resources themselves
  - Particularly natural when resources treated as objects: authorisation is a part of that object's call

Data Integrity

- Tampering: has the message been modified en route?
  - Sign messages with digital signatures using a public key method to detect any tampering with the data
  - Message can also be strongly encrypted without signing
  - Recovery is hard, don't trust message, alert sender if address is intact
- Distributed computing: can we trust unknown peers performing our computing correctly?
  - Duplicate computation, or attach a test run

Security Issues in Structured P2P Systems

- Collective surveys of P2P security
  - Androutsellis-Theotokis and Spinellis: A Survey of Content Distribution Technologies (chapter 5)
  - Dan S. Wallach: A Survey of Peer-to-Peer Security Issues
- Why study issues in structured DHT-based systems rather than Gnutella or Napster-like systems?
  - They have been subject to more extensive analysis and more careful design to guarantee scalability and efficiency
  - Provide a platform for variety of service, rather than a single specific file sharing protocol

Assumptions for Modelling Security Issues

- Set of N nodes forming a structured overlay
- Constrained-collusion Byzantine failure model
  - Fraction f $(0 \leq f < 1)$ of nodes are faulty
  - Faulty nodes can behave arbitrarily and they may not all necessarily be operating as a single conspiracy
  - Faulty nodes partitioned into independent coalitions, bounded by cN $(1/N \leq c \leq f)$. When $c = f$, all faulty nodes may collude with each other to cause most damage
- Possible to model different failure scenarios

Secure Storage

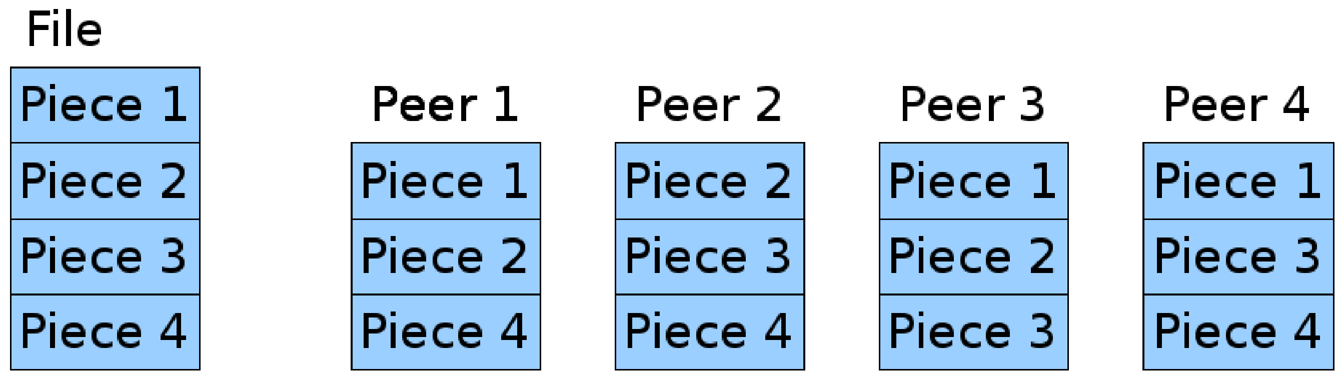
- P2P provides flexible way to share files without server, but peers may be less reliable (home computers, etc.) and suffer from censorship
- The following cryptographic algorithms and protocols provide security measures for published and stored content in a distributed P2P network
  - Also suitable for any other possible environments
  - Note: the actual concepts are clarified here, yet the algorithm details are still more complex

Self-certifying data

- Inserting node calculates hash value for target file from the content, which then becomes the key
- Node retrieves file by the key and verifies the content integrity by calculating the hash value from content
- Similar method used in PAST and Freenet
- Requires a global hash function by all nodes and using a hash-derived signature as a file's unique access key

Information Dispersal Algorithm (IDA)

- By Michael Rabin, commonly used (e.g., in JXTA)
- Disperse a file over many peers such that only a small number of peers are required to rebuild the file
- File encoded into *m* blocks that any *n* is sufficient to recover the data *(m < n)*



Basic idea in IDA: retrieve pieces from any other peer to recover the file

Secret Sharing Scheme

- The scheme helps to avoid single point of failure during the key (i.e., the secret) maintenance
  - E.g., encrypting content (file) with the key
- Key is split to *n* parts and distributed to participants, recovering the key requires *k* members of *n* *(k < n)*
- Would require eliminating *n-k-1* nodes holding the key shares to make the file inaccessible
- Shamir's Secret Sharing Scheme one popular scheme
  - Mathematical basis in the original paper How to share a secret by Shamir

Anonymous Cryptographic Relays

- *Publisher* selects several *forwarders* and sends to them via an anonymous connection shares of an encrypted file (all are peer nodes in the network)
  - Encode file pieces by Shamir's secret sharing scheme
- *Forwarders* selects other nodes as *storers* for the shares, which, in turn, will store the data
- Once all shares are stored, p*ublisher* destroys originals and announces the file name and list of *forwarders*

Anonymous Cryptographic Relays Cont'd

- Retrieving files
  - Contact f*orwarders,* which contact random servers to act as decryptors for the addresses of the *storers*
  - F*orwarders* can contact *storers* and request the shares
- Decrypting the storers' addresses (i.e., anonymous connection system) can be based on *Onion Routing*
  - Hides and protects the sender and receiver by applying sequence of proxies

Anonymous Cryptographic Relays Cont'd

- Described features can ensure, that the nodes containing the data cannot be seen at the insertion or retrieving the file
- Forwarders are visible to attackers, but compromising them reveals no data or the addresses of the storers
- Anonymity via a routing scheme, like onion routing

Routing Issues

- Best-effort service to delivery a message to a replica root associated with a given key
  - Faulty nodes can corrupt overlay-level communication
  - Cryptographic methods to authenticate objects cannot prevent malicious nodes to corrupt, delete or deny access
- Secure routing primitive requires:
  - Securely assigning node Ids
  - Securely maintaining routing tables
  - Securely routing messages

Securely Assigning Node Ids

- Ensure that attacker cannot choose the value of Id
  - For example, in CAN node itself picks a random position
  - Otherwise a coalition could concentrate near a certain key, controlling all the replica roots for the document
  - Coalition trying to maximise changes of appearing in a victim node's routing table; censoring, tampering
- CA producing cryptographic node Id certificates
  - Server maps random Id to new node's public key
  - Nodes can verify each others certificates

Sybil Attack

- Coalition of hostile nodes might try to get a large number of identities (node Ids)
  - Even if certificated node Ids
- Disproportionate control over the network if random node Ids largely collected
- Moderate the *rate* at which Ids are given out
  - Charging money or external (real-life) authentication
- Decentralised assigning of secure Ids open question

Robust Routing

- Assume that attacker controls fraction *f* of the legitimate nodes
  - Also routing tables have fraction *f* of malicious nodes
- Locality-based attacks
  - Pastry tries to fill node's routing table with “local” nodes (i.e., low latency, bandwidth)
  - Constrained routing set the table according to Id space
  - Trade-off between locality and performance
- Multiple, redundant routes between the end points

Ejecting Misbehaving Nodes

- When $f \leq 30\%$, Pastry is measured to be able to route successfully with a 99.9% probability when source sends the message first to all of its neighbours
- How to handle after that? Possible to detect and *remove* malicious nodes actively?
  - How to *prove* accusation, that another node is cheating?
  - Routing layer is difficult: dropping messages or pretending inability to route message forward can also be explained by failures of underlying Internet fabric

Fairness and Incentives in P2P Systems

- Performance and availability relies to a large extent on the voluntary participation of its users
- Necessary to employ mechanisms that provide incentives and stimulate cooperative behaviour between the users
    - Absence of a robust and secure system for making and accepting anonymous micropayments
    - Difficult to create such a system in transient population
- A notion of accountability for actions performed

Free-Riding

- An example of uncooperative behaviour
  - Users only consume resources without contributing any
- This can be interpreted as a manifestation of the "Tragedy of the Commons"
  - Argues that people tend to abuse shared resources that they do not have to pay for in some way
- Specific solutions exist, for example, in BitTorrent the *tit-for-tat* game strategy
- Could be solved with incentives and trust schemes?

Trust in P2P

- In cryptographic systems, trust usually means the authenticity of an entity
- For P2P it has broader implications: trust is the confidence that a peer has to ensure that it will be treated fairly when interacting with another peer
- Decentralisation means that building trust relies on peer collaboration
  - How to persuade (most of) peers to cooperate in an open and unregulated environment, such as the Internet?

Trust in P2P Cont'd

- Trust example: searching based on keywords
  - Strong hashing can guarantee valid file content, but what about spoofing the keyword search results?
  - Slipping in Trojans and back-doors, degrade the service
  - Also, recording industry deploying “decoy” files
- One solution would be to form a notion of reputation
  - Similar to Google's PageRank technology
  - The research questions follow: how peers can rank each other? What are the alternatives?

Properties of P2P Trust Schemes

- Trust is based on some types of feedback from peers
  - Based on past experience of cooperation (good or bad)
  - Should consider more than one type: mere past positive feedback allows colluding peers to raise each other's reputation falsely, but using only negative past feedback gives newly joined peers unfair benefit
- Authentication and non-repudiation
  - Confirming which peer gave feedback and being able to validate that the sender and receiver, in fact, were the claimed peers (requirements towards accountability)

Properties of P2P Trust Schemes Cont'd

- Communication and storage costs
  - While using a trust scheme, peer needs to query the participating peer's trust value before decision to trust
  - More accurate knowledge requires gathering more data
  - Without centralised service, there's a trade-off between querying other peers about the information and accuracy
  - Also, more trust data requires more storage and computing
- Cost questions always tightly related to scalability
- Anonymity: protecting the feedback giving peer

Attacks Against Trust Schemes

- Sybil attack
  - malicious peers gaining multiple identities weakening idea of peer contributing only once (e.g., voting systems)
- Denial of service
  - Transaction-based systems: flooding numerous fake feedback trough fake transactions
  - Attacking negative feedback providers outside the overlay
- False accusation
  - Providing false accusations/reports against innocent peers

Attacks Against Trust Schemes Cont'd

- Context-based attacks
  - What is important-rated transaction? Most existing systems cannot consider semantics of transactions
  - Honest cooperation in many small-valued transactions, targeting cheating in large-valued transactions
- Strategic dynamic personality based attacks
  - Building positive feedback (e.g., eBay-like reputation)
  - Cheating or oscillating between building and milking the reputation

General Solution Schemes

- Trust-based incentive mechanisms
  - Engage a transaction based on whether you trust on the other party
  - Reputation mechanisms belong in this category
- Trade-based incentive mechanisms
  - A party offering some service to another is explicitly remunerated, either directly or indirectly
  - Mainly represented by various micropayment (currency-based) mechanisms and resource trading schemes

Reputation Mechanisms

- “Online word-of-mouth communities”, individuals sharing opinions about other individuals
- P2P system must distribute the reputation information
- Goal: take locally generated reputation information resulted by an interaction of peers and spread it through the network as a global rating for peers
  - Security and availability of this information
- Reputation captured through metrics and tools for measuring behavioural characteristics of users

Examples of Reputation Mechanisms

- EigenTrust algorithm by Kamvar et al.
  - Global reputation values based on peers' history of uploads, calculated from local values assigned by other peers, weighted by the global reputation of assigning peers
- Various simple approaches
  - Kazaa calculates user participation level and rewards by giving higher priority to requests (file transfer queue)
  - An utility function estimating peer's usefulness based on the amount and popularity of content stored by peer

Micropayment Mechanisms

- One of earliest centralised P2P micropayment: MojoNation
  - A currency is gained by offering disk space, bandwidth, or CPU cycles for obtaining access to distributed storage space, trusted third party to ensure honest transactions
- Searching example: querying requires a token
  - Receiving a query adds a token to the user
  - What if a node has tokens left and decides to refuse query?
  - Requires more widely replicated data and token's validity

Resource Trading Schemes

- Distributed auditing
  - Guaranteeing a fair storage usage for all peers
  - Barter economy of disk space: trade use of local disk for the use of others remote storage
  - Nodes publish and sign logs: local list of files they are storing on behalf of others and remote list of files that other are holding on behalf of the local node
  - Log file integrity managed by random auditions, which reveals tampering the log when comparing to other results

Trust Conclusions

- Neither reputation-based or trade-based can solve all problems, select suitability according to the case
- Currency-based trading avoids “double coincidence of wants”, but has higher complexity
- Trade-based trading works well in homogenous environment, where peers have similar consumptions
  - Inappropriate for highly ad hoc environments
  - High imbalance in contribution of popular objects leads may lead to wasting exchanged resources

Trust Conclusions Cont'd

- Reputation-based trust schemes are the mainstream
  - Similarity between the chaos of a P2P system and the disorder in a human society at its initialisation stage
  - Flexibility: can be used for any P2P, avoids the limitations on trade-based schemes
- Main weakness: becomes operational only after a node has misbehaved
  - Using another method in boostrapping stage (new peer)
- Nearly every system uses its own variant of a scheme

Related Matters

- Anonymity: providing privacy, confidentiality and censorship resistance (E.g., Freenet)
  - In content distribution anonymity can refer to the author of content, identify of a node storing the content itself, or the details of a query for retrieval of the content
- Deniability: user's ability to deny knowledge of content stored and/or content being transferred
  - Increasing the degree of freedom while decreasing it for controllability

Lecture Conclusions

- Decentralisation and Internet makes security hard
  - Many popular file sharing protocols lack of it, structured systems have better security basis
  - Conflicting aims; search efficiency and anonymity
- Security is followed by trust and fairness
  - Building trust between participants is essential for business and production environments, and sensitive data
  - Methods that aim to guarantee, that the system can performs well and avoids the “tragedy of the common”

Lecture Conclusions Cont'd

- Many techniques ranging from cryptography through redundant routing to economic methods
  - Due to diversity of P2P system design and usage, there is a corresponding diversity of security, fairness and trust solutions applied to the problems
- Security and trust issues becomes even more critical when P2P is applied to e-business, and involves sensitive information

Considerations

- Security level in P2P systems studied so far?
  - E.g. Gnutella, Freenet, Bittorrent, DHT-based, also others, like SETI@home and Skype
- First, think about the theoretical basis, what are their aims, do they even try to provide any security?
- Why to break into a regular desktop computer?
  - Have you received (i.e., are you aware of) any malicious material through a P2P application?
- Do those systems provide any trust or fairness?