Peer-to-Peer Systems and Security

Security

Christian Grothoff

Technische Universität München

April 14, 2013

"It's not good enough to have a system where everyone (using the system) must be trusted, it must also be made robust against insiders!" – Robert Morris, former Chief Scientist of the US National Security Agency (NSA)

Peer-to-Peer Systems and Security

- In a *pure* P2P system, everyone is an insider

$\Rightarrow$ No other peer can be trusted — for anything

$\Rightarrow$ No certificate authorities, trust anchors, etc.

$\Rightarrow$ Achieving any kind of security is very hard!

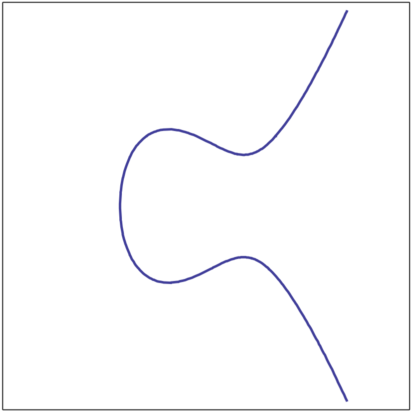
Cryptographic Primitives

- Random number generation
- Hashing
- Symmetric encryption
- Asymmetric encryption

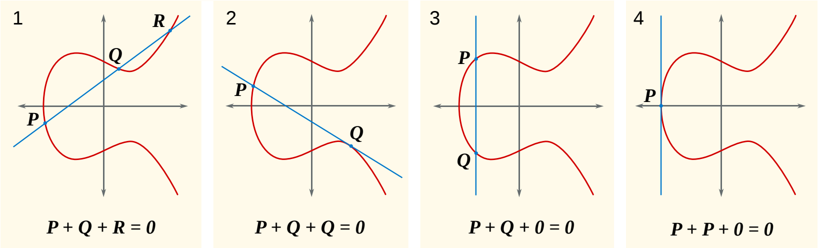
Look at `gnunet_crypto_lib.h` if you need any of those.

Detour: Elliptic Curves

- Modern Public-Key crypto
- $y^2 = x^3 + ax + b$
- $0 = (\infty, \infty)$



Elliptic Curve Point Addition



Elliptic Curve Cryptography

- If we can calculate $P + P$, we can calculate dP for $d \in \mathbb{N}$
- Pick discrete curve over $\mathbb{F}_p$
- Find generator G of order n (n minimal such that $nG = 0$)
- (p, a, b, G, n) identifies the curve
- $d \in \mathbb{F}_n$ is the private key
- $Q := dG$ is the public key
- Can now do DH and DSA (called ECDH and ECDSA)

Security Goals

- Availability
- Confidentiality
- Integrity
- Authenticity

P2P Authentication

How to authenticate in a pure P2P system?

P2P Authentication

How to authenticate in a pure P2P system?

- Public key $\equiv$ identity ($ID_x := H(PK_x)$)
- Alice can then sign her messages: $A, PK_A, S_A(M)$

Such identifiers are called “cryptographic identifiers” (or self-certifying identifiers).

Boyd's Theorem

Can we use traditional identifiers (i.e. names) in an open P2P system?

Boyd's Theorem

Can we use traditional identifiers (i.e. names) in an open P2P system?

Theorem (Boyd's Theorem I)

"Suppose that a user has either a confidentiality channel to her, or an authentication channel from her, at some state of the system. Then in the previous state of the system such a channel must also exist. By an inductive argument, such a channel exists at all previous states."

Boyd's Theorem

Can we use traditional identifiers (i.e. names) in an open P2P system?

Theorem (Boyd's Theorem I)

"Suppose that a user has either a confidentiality channel to her, or an authentication channel from her, at some state of the system. Then in the previous state of the system such a channel must also exist. By an inductive argument, such a channel exists at all previous states."

Thus, no secure channels may be formed between any users who do not already possess secret or shared keys.

Boyd's Theorem

Theorem (Boyd's Theorem II)

"Secure communication between any two users may be established by a sequence of secure key transfers if there is a trusted chain from each one to the other."

Boyd's Theorem

Theorem (Boyd's Theorem II)

"Secure communication between any two users may be established by a sequence of secure key transfers if there is a trusted chain from each one to the other."

$\Rightarrow$ No secure, in-system authentication without trusted third parties or prior contacts.

Authentication without Authorities

- Add out-of-band mechanisms (i.e. GNUnet F2F mode)
- Use social properties (security graph $\Leftrightarrow$ social network graph)
- Use network properties (i.e. IP address)
- Key continuity / baby duck — assume first contact to be secure (i.e. `ssh`)
- Group decisions
- ...

Zfone Authentication (ZRTP) [3]

Idea: combine human interaction proof and baby duck approach:

- A and B perform Diffie-Hellman exchange
- Keying material from previous sessions is used (duckling)
- Short Authentication String (SAS) is generated (hash of DH numbers)
- Both users read the SAS to each other, recognize voice

A man-in-the-middle attacker usually needs to intercept and change the Diffie-Hellman numbers to perform the attack on the initial exchange.
$\Rightarrow$ ZRTP foils standard man-in-the-middle attack.

Trust vs. Authentication

In open P2P networks, we care less about who operates a peer.
We want to know if a peer will behave:

- Will a peer follow the protocol?
- Will a peer share resources (such as files)?

Trust vs. Authentication

In open P2P networks, we care less about who operates a peer.
We want to know if a peer will behave:

- Will a peer follow the protocol?
- Will a peer share resources (such as files)?

We can never be **sure** about a peer ...

- keeping our secrets once we expose them
- being our “friend”

Trust

The term "trust" can be used with slightly different meanings:

- A **trusted party** is a party that we trust completely for particular operations (within the technical system) — we assume correct behaviour with respect to protocol and data usage.
- Trust can also be used to imply **authorization** — we trust a party (such as a human or organization) with important or private information.

A related issue is **revocation**, the removal of authorization or the withdrawing of the special trusted party status from some party.

Incentives

- Incentives are mechanisms to make a peer cooperate by giving benefits

$\Rightarrow$ BitTorrent's tit-for-tat gives uploaders increased download rates

Reputation

- Trust into a service or peer based on experience or a-priori knowledge
- Global: reputation is system-wide
- Local: each node locally computes a reputation value for each other node
- GNUnet file-sharing's "respect" in other peers is a local reputation

Reputation

- Trust into a service or peer based on experience or a-priori knowledge
- Global: reputation is system-wide
- Local: each node locally computes a reputation value for each other node
- GNUnet file-sharing's "respect" in other peers is a local reputation

Reputation requires **observation**, **evaluation**, **storage** and **predictability**.

Attacks on Reputation

- Time-dependency — attacker may behave well for a while, then change behavior (Ebay attack)
- Whitewashing — badly-rated peer leaves and returns with new "innocent" identity
- Collusion of attackers — attackers give each other good ratings

Sybil Attack

Background:

- Ancient Greece: Sybils were prophetesses that phrophesized under the devine influence of a deity. Note: At the time of prophecy not the person but a god was speaking through the lips of the sybil.
- 1973: Flora Rheta Schreiber published a book Sybil about a woman with 16 separate personalities.

Sybil Attack

Background:

- Ancient Greece: Sybils were prophetesses that phrophesized under the devine influence of a deity. Note: At the time of prophecy not the person but a god was speaking through the lips of the sybil.
- 1973: Flora Rheta Schreiber published a book Sybil about a woman with 16 separate personalities.

The Sybil Attack [1]:

- Insert a node multiple times into a network, each time with a different identity
- Position a node for next step on attack:
  - Attack connectivity of the network
  - Attack replica set
  - In case of majority votes, be the majority.

Defending against Sybil Attacks

- Use authentication with trusted party that limits identity creation
- Use "external" identities (IP address, MAC, e-mail)
- Use "expensive" identities (solve computational puzzles, require payment)

Douceur: Without trusted authority to certify identities, no realistic approach exists to completely stop the Sybil attack.

Sybil Defense: The Bootstrap Graph

Assumptions:

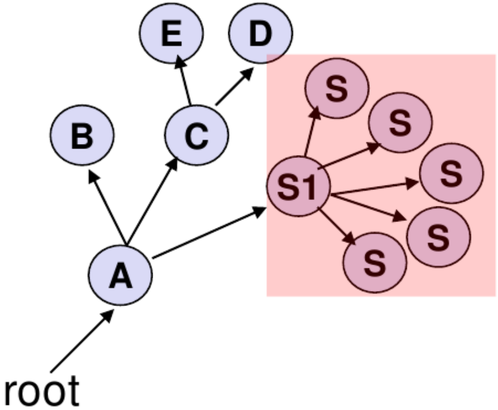
- The first Sybil node enters via an arbitrary bootstrap node
- The rest of the nodes will join via another Sybil node

Sybil Defense: The Bootstrap Graph

Assumptions:

- The first Sybil node enters via an arbitrary bootstrap node
- The rest of the nodes will join via another Sybil node

In the bootstrap tree, nodes are a child of the node they used to bootstrap:



Sybil Defense: Bootstrap Graph

Idea: when selecting peers, use nodes from different subtrees in the bootstrap graph.

Sybil Defense: Bootstrap Graph

Idea: when selecting peers, use nodes from different subtrees in the bootstrap graph.

Assumptions:

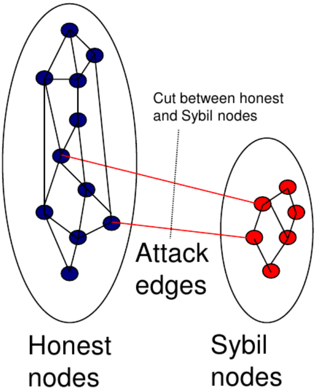
- The first Sybil node enters via an arbitrary bootstrap node
- The rest of the nodes will join via another Sybil node

⇒ Bootstrap node must enforce access control policies, i.e. based on social relationships

Sybil Defense: SybilGuard [4]

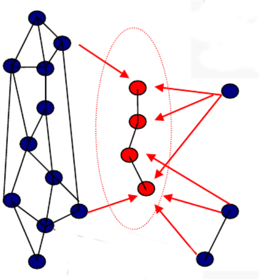
- Sybil nodes primarily know each other

⇒ Small cut between subgraph of honest nodes and subgraph of Sybils.



Eclipse Attack: Goal

- Separate a node or group of nodes from the rest of the network
- isolate peers (DoS, surveillance) or isolate data (censorship)



Eclipse Attack: Techniques

- Use Sybil attack to increase number of malicious nodes
- Take over routing tables, peer discovery

⇒ Details depend on overlay structure

Defenses

- Large number of connections
- Replication
- Diverse neighbour selection (different IP subnets, geographic locations)
- Aggressive discovery ("continuous" bootstrap)
- Audit neighbour behaviour (if possible)
- Prefer long-lived connections / old peers

Poisoning Attacks

Peers can provide false information:

- wrong routing tables
- wrong meta data
- wrong index information
- wrong performance measurements

Timing Attacks [2]

Peers can:

- measure latency to determine origin of data
- delay messages
- send messages using particular timing patterns to aid correlation
- include wrong timestamps (or just have the wrong time set...)

Questions?

?

References

John Douceur.
The Sybil Attack.
In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.

Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright.
Timing attacks in low-latency mix-based systems.
In *Proceedings of Financial Cryptography (FC '04)*, pages 251–265, February 2004.

Laurianne McLaughlin.
Philip zimmermann on what's next after pgp.
IEEE Security & Privacy, 4(1):10–13, 2006.

Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman.
Sybilguard: defending against sybil attacks via social networks.
SIGCOMM Comput. Commun. Rev., 36(4):267–278, August 2006.