PROFILES: PEER-TO-PEER BEYOND FILE SHARING

Distributed security for P2P networks

Leonardo Maccari

Universitá di Firenze
Laboratorio di Reti e Telecomunicazioni

Outline

1. Distributed authentication
   - Public/Private key schemes
   - Shared key schemes
   - Applications
2. Packet filtering and classification for distributed networks
   - Bloom Filters
   - Applications

Authentication/access control techniques

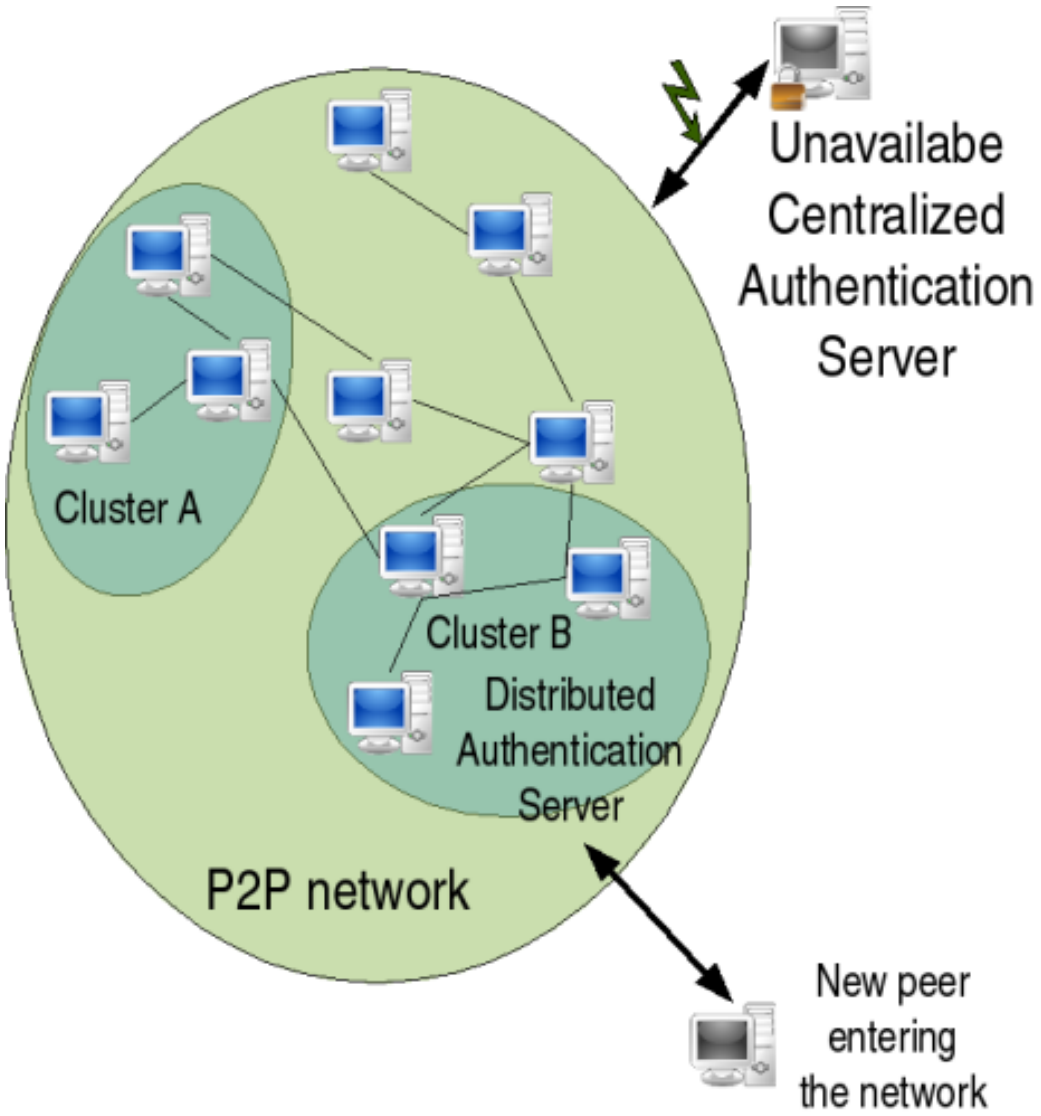
- Authentication: The assurance that the communicating entity is the one that it claims to be.
  - Peer Entity Authentication: provides confidence in the identity of the entities connected
  - Data Origin Authentication: provides assurance that the source of received data is as claimed
- Access control: The prevention of unauthorized use of a resource.

Authentication techniques are commonly used to provide access control

Authentication techniques

- Asymmetric techniques: RSA, DSA, and other techniques based on private/public key cryptography.
- Symmetric techniques: Shared key mechanisms, MAC/MIC functions (Message Authentication/Integrity Codes) etc. . .

Scenario


Unavailabe
Centralized
Authentication
Server
Cluster A
Cluster B
Distributed
Authentication
Server
P2P network
New peer
entering
the network

Distributed authentication

- Distributed authentication is the technique used by a group of collaborative nodes that want to have some trust relation, without the use of any trusted third party authority.
- There exist in literature[1], several variants of classical authentication algorithms generalized to be used in a distributed fashion.
  - Distributed RSA
  - Distributed DSA
  - ...
- Each of the proposed public-key based techniques have in common the use of Public/Private cryptographic key schemes.

[1] Saxena, N.T. and Yi, G.J.H.: Access Control in Ad Hoc Groups, International Workshop on Hot Topics in Peer-to-Peer Systems, 2004.

Example: distributed RSA

- The basic idea behind this techniques is that each of the participants (let's say M peers) is in possession of a *share* of a private key
- the collection of a number M of shares of the secret can be used to produce a valid signature
- the scheme is robust against insider enemies, it includes instruments to verify that the partecipants are generating correct signature shares.
- if one of the shares is lost, or the verification of the signatures shows that a node is cheating, each of the share can be reconstructed by a cooperation among a subset of K<M nodes

Shamir's shared secret

Shamir proposed a simple way to share a secret between a group of peers.

Setup

- A polynomial P(x) of degree K-1 is defined, such as the secret is $P_0 = P(0)$
- each participant receives a couple (x_i, y_i) where $P(x_i) = y_i$

Evaluation

- If at least K peers join, they are able to recreate the secret
- It a coalition of X<K peers behaves maliciously they are unable to derive any information on the secret.
- The scheme is practical (easy to add peers to the original group), but once the secret is revealed, the *dealer*, the peer who collected the shares has control over the polynomial.

Multiple polynomials

In the area of Wireless Sensor Networks (WSN) there is no possibility of using public key cryptography (for hardware limitations) so polynomials are gaining interest as a substitute.

- Can the simple Shamir' Scheme be extended to support re-use of the secret?
  - for WSN we are developing an access control protocol based on the use of multiple polynomials and hash functions
  - the protocol should be able to let a subset of the nodes authenticate a new node, without any central authority
  - security should be guaranteed up to the compromission of a pre-determined number of nodes

Another scheme: Blundo's dynamic conferences

Can this scheme be extended to be used for access control in a peer group? If this was possible, a lightweight distributed access control protocol could substitute the public/private scheme.

- Blundo's key distribution scheme:
  - Let P(x,y) a polynomial so that given x_0, y_0 the value of P() is the same for any permutation of the two values (i.e. P(x_0, y_0) = P(y_0, x_0))
  - Each user *i* receives a polynomial G(z) = P(i,y)
  - If the users 1 nad 2 want to set up a shared secret:
    - User 1 computes $G(2) = P(1, 2)$
    - User 2 computes $G(1) = P(2, 1)$
    - Since $P(2, 1) = P(1, 2)$, the two parties have a shared secret. This can be generalized to a $P(x_0, x_1, x_2 ... x_n)$ polynomial to have a group of n peers.

Applications: private group of peers

- These schemes could be studied to create communities of peers with cryptographic protection of privacy, authentication and access control. In particular:
  - Shamir/Blundo scheme seems suitable for the creation of autonomic groups of peers
  - Threshold RSA/DSA could be used to set up a distributed CA, and allow decentralized authentication based on public/private key (applicable for example to Trusted Computing devices for user identification against freeriding)

Reputation and trusted computing

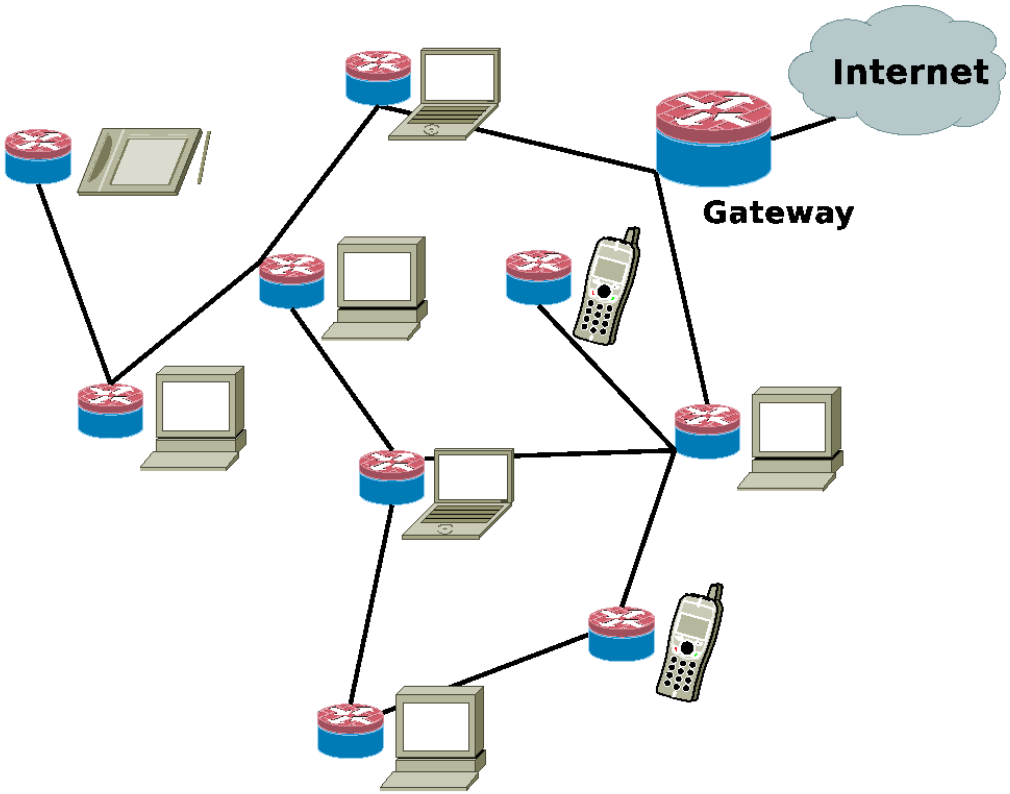
- Freerider: a node that tries to exploit the resources of the network without cooperating
- in a reputation system, a freerider quickly gains a bad reputation, but he might be able to log-out and relog, to *clear* its reputation
- trusted computing (TC) is based on the use of hard-coded public/private keys, that can not be changed by the user
- TC introduces privacy problems, becouse the actions of a user can be directly connected to a hardware device
- the use of a distributed authentication authority could mitigate this problem.

Firewalling

A firewall is a hardware or software device which is configured to permit, deny or proxy data through a computer network which has different levels of trust.

- Normally firewalling is applied to bastion hosts at the entrance/exit of private networks.
- Firewalling in wireless distributed networks (mesh networks) is actually unapplied, but it would be extremely useful.
  - If a mesh network is made of AP, each of them having a set of clients, if a clients start transmitting unauthorized traffic (i.e. SPAM, or DOS) the gateway to the Internet will filter it out, but. . .
  - . . . the main concern about mesh networking is stability of the network itself, which is usually resource constrained.
  - so if the traffic gets to the gateway, then the damage is already done.

Mesh Network



Firewalling

- How is firewalling done? Normally a firewall is configured with a set of rules of the type:
  - if IP Source == A, IP dest == B, TCP source port == C, TCP dest port == D $\rightarrow$ ACCEPT the packet
- Each rule is evaluated separately for each packet, until one doesn't match (the default is DROP)
- The more fine grained the rules are, the more flexible the firewall is
- if a distributed network is composed of N nodes, and each node can communicate to each other over K TCP ports, we have $2 * k * N^2$ rules!
- since the complexity of search is linear with the number of rules the evaluation time of each packet grows with the number of rules

Bloom Filters

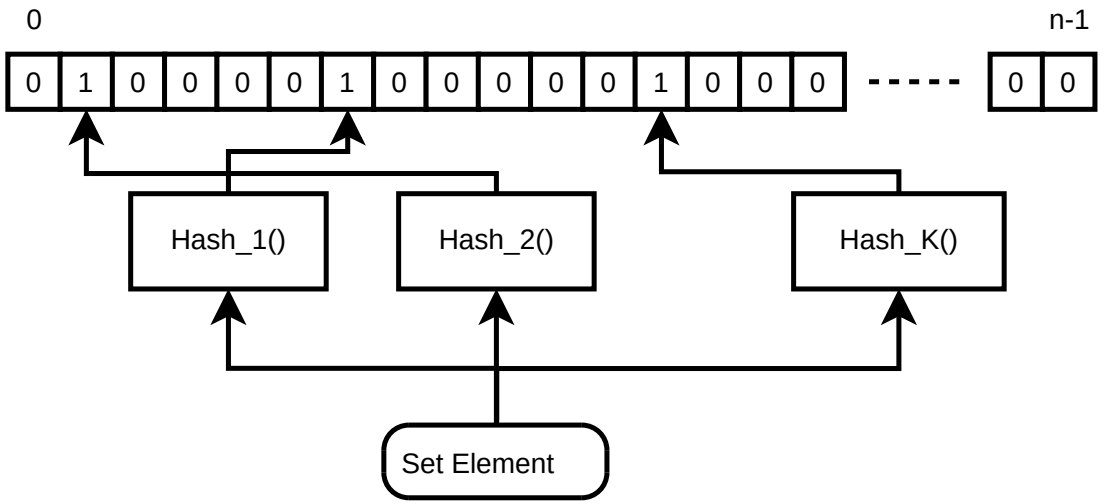
We studied the possibility of using a compact data structure, Bloom filters[2]

- a space-efficient probabilistic data structure that is used to test whether an element is a member of a set
- false positives are possible, but false negatives are not
- the probability of false positives can be parametrized.

[2] A. Broder and M. Mitzenmacher: Network Applications of Bloom Filters: A Survey, In Proc. of Allerton Conference, 2002.

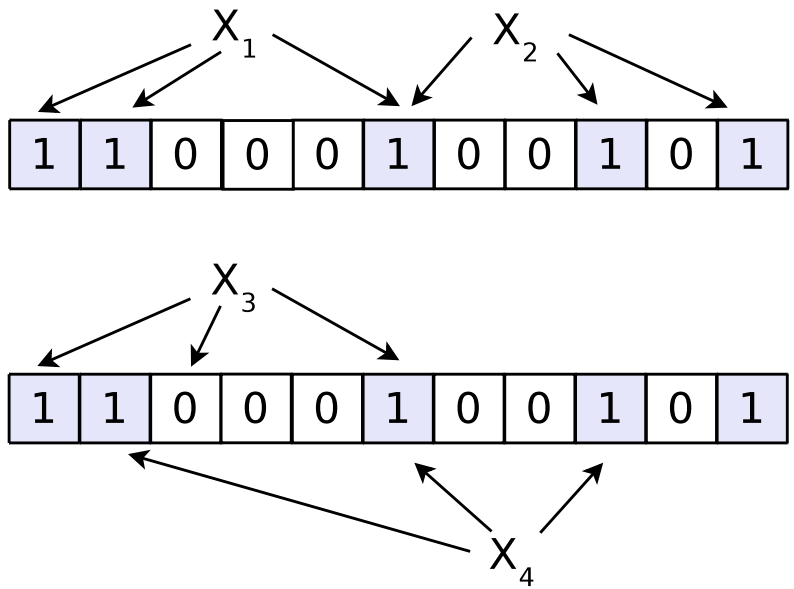
Populating Bloom Filters

- An empty Bloom filter is a bit array of n bits, all set to 0.
- There must also be k different hash functions defined, each of which maps a key value to one of the m array positions, each function returns a binary string of length $log_2 n$ so that it can be used as index for the filter.
- Each element of the set is hashed with each of the functions and determines a mask of bits to be set to 1 in the empty filter.



Querying Bloom Filters

- Once the filter is populated, to make a query, the element to be tested is hashed and the generated mask is compared with the BF.
- if all the corresponding bits are set to 1, the element belongs to the set.
- if any of the bit of the mask is set to 0 in the BF, then the element doesn't belong to the set (no false negatives).
- false positives?



- for this same reason Bloom filters do not support deletion of elements once the filter is populated.

Proprieties

- The OR of two filters is a filter representing the union of the set
- the AND of two filters is a filter representing the interception of the set
- the hash functions can be calculated in parallel, which is useful for hardware implementations

Bloom Filters for Firewalling

- elements of the set are of the type:
  - $s_i = \{IP_{src}, IP_{dst}, Port_{src}, Port_{dst}\}$
- for each received packet the hashes are computed and the query is done
- there can be only false positives (packets are allowed that should be dropped)
- no false negatives (allowed packets that will be dropped)
- computing time constant for positive queries (calculation of k hash functions), less for negative queries (at most k hash functions)

Bloom Filters for Firewalling

- The probability of having a false positive if the original set is composed of m elements, BF is of length n and K hashes are used is given by:

$$f = (1 - (1 - 1/n)^{-Km})^K \simeq (1 - e^{-Kn/m})^K \tag{1}$$

- f is minimized if $K = ln(2) * m/n$ which gives:

$$f = (0.5)^K = (0.6185)^{m/n} \tag{2}$$

- If we set $f = 0.1\%$ we have:

$$\begin{cases} m/n = \lceil log_{0.6185}(0.001) \rceil = 15 \\ K = \lfloor log_{1/2}(0.6185^{20}) \rfloor = 10 \end{cases} \tag{3}$$

- roughly, to have a 0.1% probability of false positives 15 bits per element of the set should be used and 10 hash functions are needed

Bloom filters evolutions

Many Bloom filter variants have appeared:

- Counting BF (supporting deletion)
- Compressed BF (optimization for compressed size)
- Generalized BF (supporting complex queries)
- d-left BF

Simple application to Firewalling

- build a set S of values $s_i = \{IP_{src}, IP_{dst}, Port_{src}, Port_{dst}\}$ corresponding to allowed packets
- load an appropriate sized BF with each element of the set
- each time a packet is received, extract its features and evaluate the presence of a positive rule.

P2P filtering

How can this be applied to P2P?

- Similarities exist between certain models of P2P networks and mesh networking
- the main difference is that in P2P networks communications may take place directly between couples or remote IP, while in mesh networks there is a strict dependency on geographical location
- research can be hop-by-hop, so that the model can be reproduced

Conclusion

Is it feasible to create close communities in P2P application networks, based on distributed firewalling and access control?