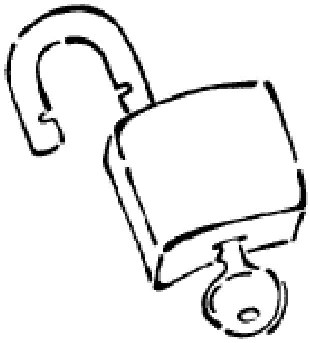
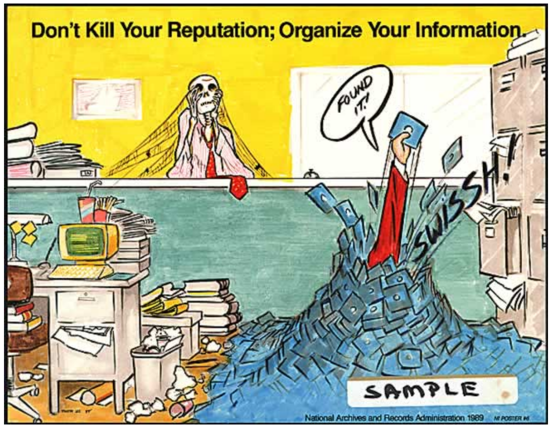
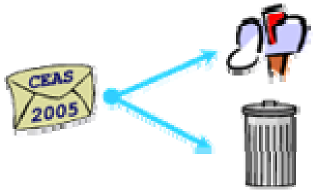
Netizen, Authentication and Reputation



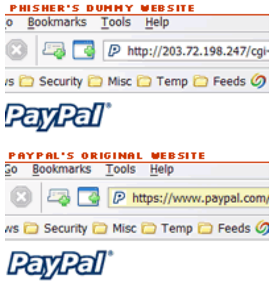

Don't Kill Your Reputation; Organize Your Information.
FOUND IT!
SWISSH!
SAMPLE
National Archives and Records Administration 1989

Shalendra Chhabra
University of California, Riverside
http://www.cs.ucr.edu/~schhabra
http://www.spam-research.com
schhabra@cs.ucr.edu

Slides at: www.cs.ucr.edu/~schhabra/ceas05.pdf



Venue: CEAS 2005, Stanford University
Thanks to Joshua Goodman, Microsoft Research

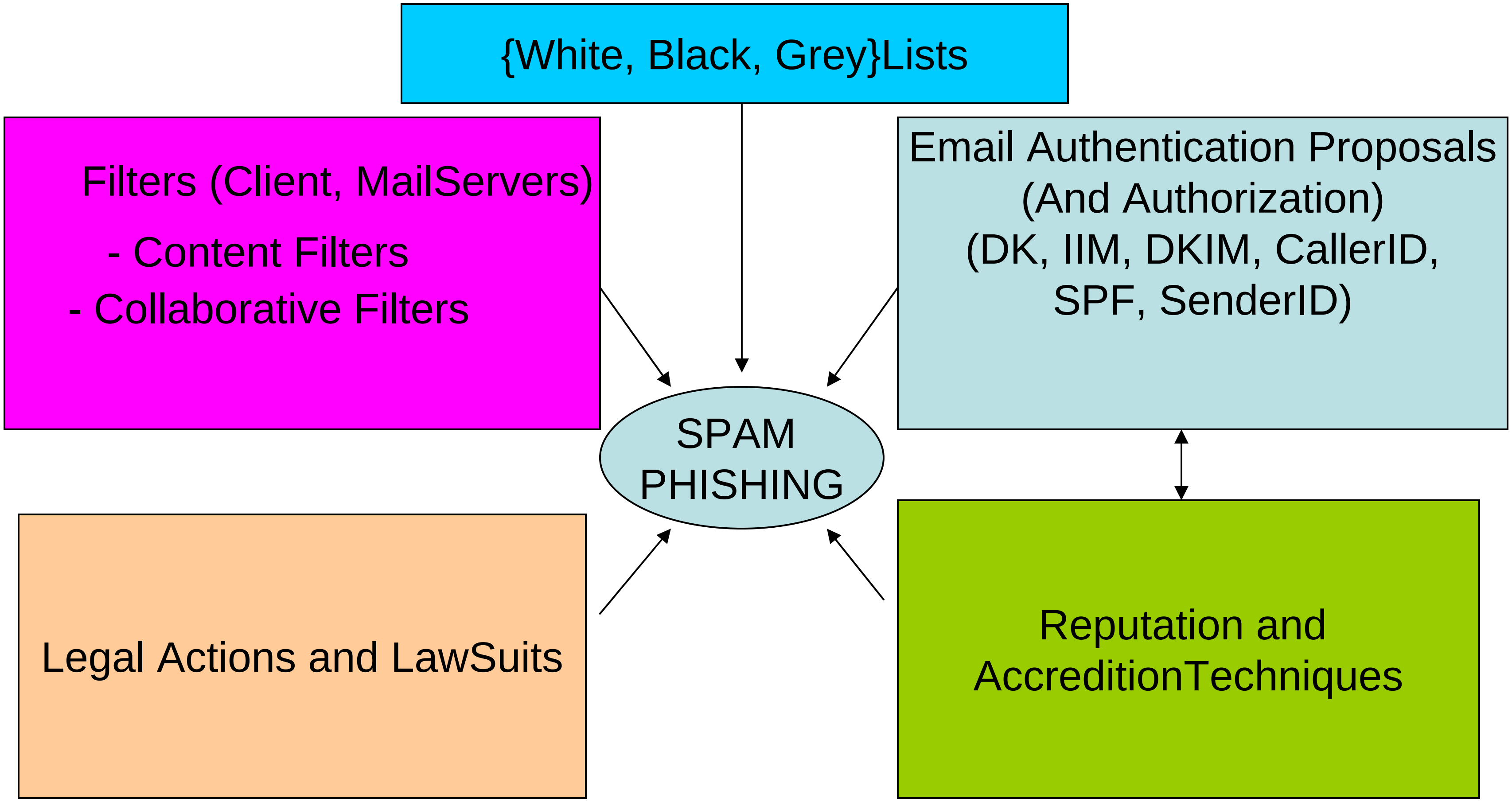


Spammers and Phishers

We Have A SwAK (Swiss Army Knife) in The Making ☺

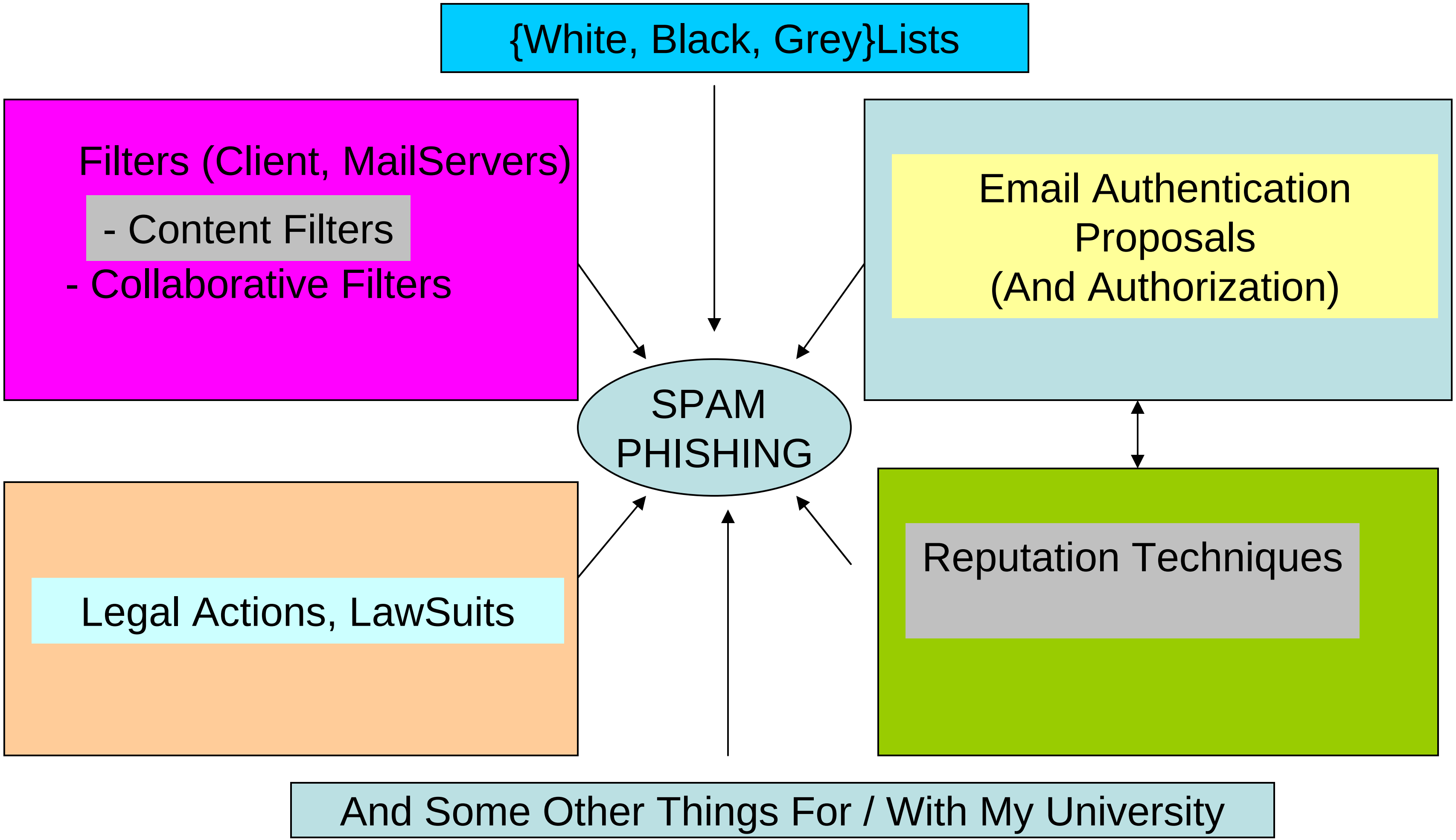
Anti Spammers

Our SWaK for Tackling Spam and Phishing


{White, Black, Grey}Lists
Filters (Client, MailServers)
- Content Filters
- Collaborative Filters
Email Authentication Proposals (And Authorization) (DK, IIM, DKIM, CallerID, SPF, SenderID)
SPAM PHISHING
Legal Actions and LawSuits
Reputation and AccreditionTechniques

Masters Thesis* (Advisor Dimitrios Gunopulos, UCR)

"Fighting Spam, Phishing and E-mail Fraud"



{White, Black, Grey}Lists

Filters (Client, MailServers)

- Content Filters
- Collaborative Filters

Email Authentication Proposals (And Authorization)

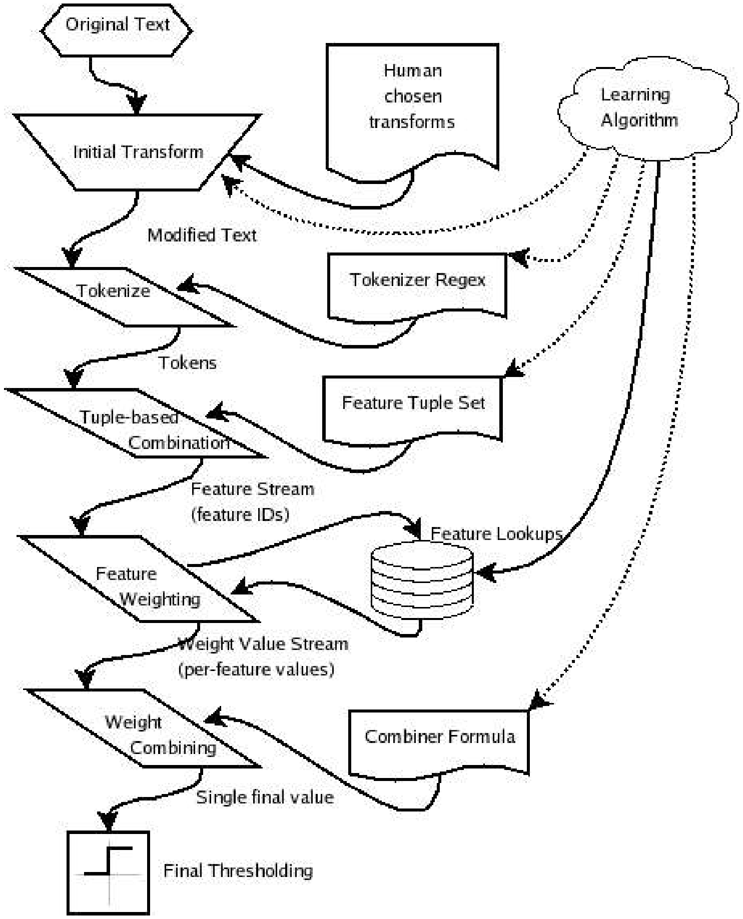
SPAM PHISHING

Legal Actions, LawSuits

Reputation Techniques

And Some Other Things For / With My University

A Unified Model of Spam Filtration MIT Spam Conference, 2005*



Authentication and Authorization

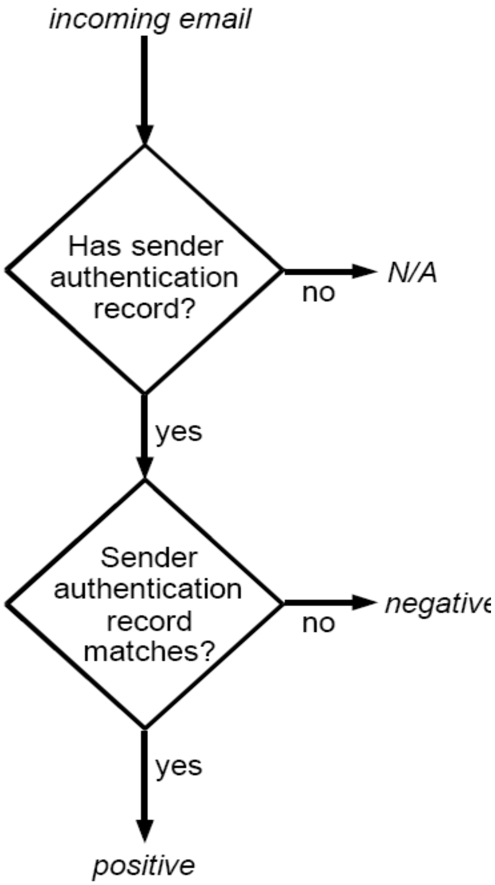
- Authentication is the process of checking or verifying an entity using some form of integrity information such as an authorization policy.
- Cisco's IIM, Yahoo's DK, now DKIM, SPF, Microsoft's CallerID now SenderID

With Email Authentication Systems What's Going to Happen Next?

- Spammers are adept at deploying sender authentication technologies for domains they are not forging
- Timeliness /reputation of domain in place
- Spammers will send from non-forged addresses (Blacklisting is the solution)

State with Email Authentication Systems *

(John Graham Cumming)


incoming email
Has sender authentication record?
no
N/A
yes
Sender authentication record matches?
no
negative
yes
positive

Forged Message or False Negative

Use Bayesian Filter to Train (State, Output) ☺

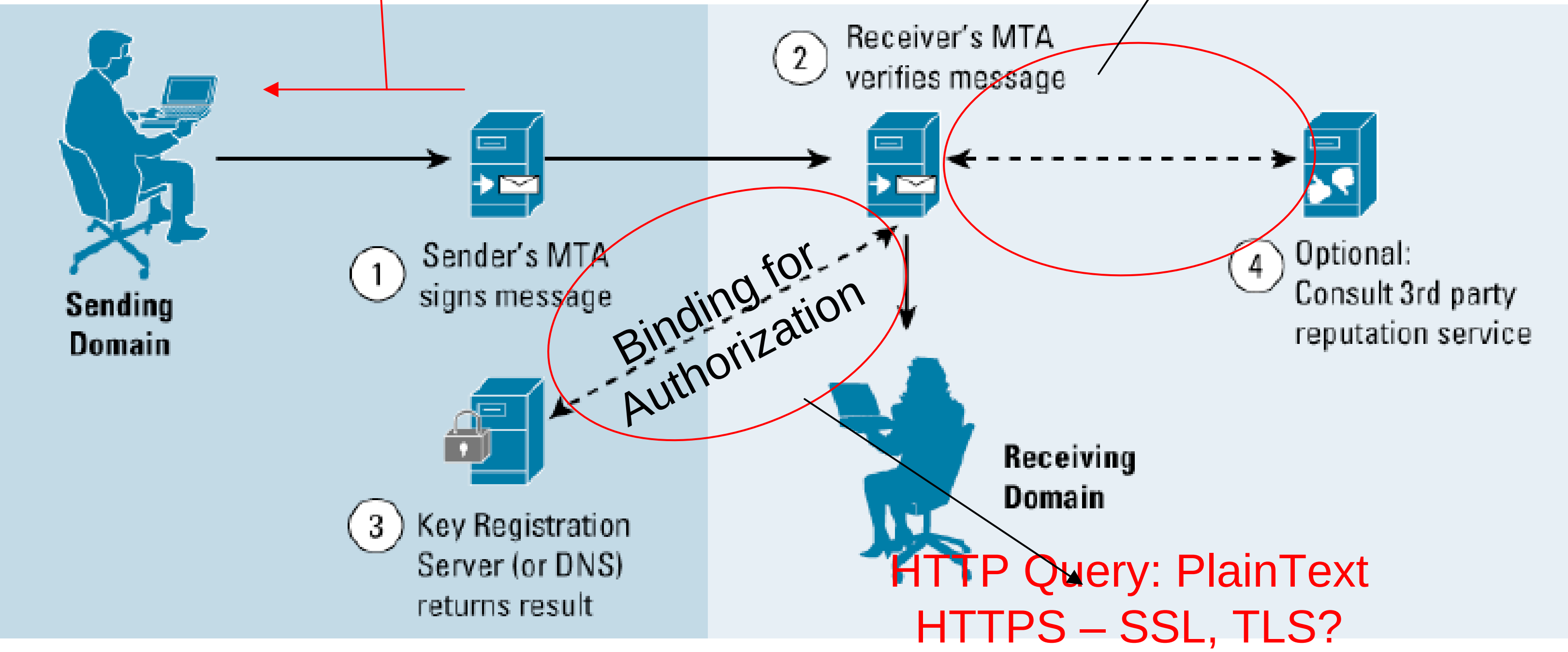
Only sure when its positive: like whitelists

Attack(s) on Cisco’s IIM (Before DKIM)

Sending Domain checks if the Source is allowed to send Mail using its Domain

Analysis of Reputation Attacks (Adapt *IDStealth, Shilling, PseudoSpoofing* and Check)

Typical Identified Internet Mail Message Flow


Sending Domain
1 Sender’s MTA signs message
2 Receiver’s MTA verifies message
3 Key Registration Server (or DNS) returns result
4 Optional: Consult 3rd party reputation service
Receiving Domain

Binding for Authorization

HTTP Query: PlainText
HTTPS – SSL, TLS?

Response Format with values not mentioned in RFC (Locally Sensitive Hash) ex: Nilsimsa Hash?

Check Possibility of These Attacks when using Third Party Reputation Services with Email Authentication Systems

- *PseduoSpoofing*: Forging great number of votes from a single node, giving them different IP addresses, and multiple IDs
- *Shilling*: Clique / Control over many participants affecting reputation
- *ID Stealth*: Malicious Agents respond in the same format as if generated from genuine servents (Challenge Response can detects this)
- *Replay Attack:* Use of Timestamps, Nonce

Reputation: Whats the Deal

- Reputation History, NewComer and Vouching Problem
- Reputation Format, Reputation Response with a Signature? *(Accountability)*
- Consistent Framework for accessing reputation required otherwise Chaos
- reputation@ironport.com

Phishing Attacks, Reputation

- Planning (Targets, Attack Methods)
- Setup (Destinations, Contacts)
- Attack (Attack Mediums via websites etc.)
- Collection (Forms, Malware, Social Engineering)
- Fraud (False Registrations) → Reputation
- Post- Attack (Destroying Evidence)

Reputation Engines and Architecture

<u>Architecture</u>

- Centralized Architecture
- Distributed Architecture Like SupRep*

<u>Reputation Computation Engines</u>

- Summation of Votes
- Bayesian Systems
- Discrete Trust Models
- Flow Models as Google's PageRank, Attack Resistant Trust Metrices (like Advogato)

*SupRep: Shalendra Chhabra etal, IEEE DEXA, 2004

Attributes, Reputation Query and Response Formats

- Issues: TCP vs UDP: Pros/Cons
- Scoring System in Reputation should be:

- ➢ Accurate for long term performance
- ➢ Should have a weight towards current participant behaviour and should reflect the score/opinion of its participants
- ➢ Should be efficient and convenient to recalculate a score quickly
- ➢ Should be robust against attacks
- ➢ Should be amenable for statistical evaluations
- ➢ Should be smooth, easy to verify if required (depends)
- ➢ Scores should imply an attribute that requestor can interpret/understand (depends upon the context)

More Design Issues… Food for Thought

- Reputation Repository
- Registration, Reputation Lookup and Update Formats
- The Reputation protocol designers should prove the protocol robust in the presence of “good, confused and bad participants”
- The protocol should allow for updates during events like entry/exit of reputation servers (if it has a distributed architecture) Ex: SupRep*

SupRup: Shalendra Chhabra etal. Italy – Spain Summer 2004

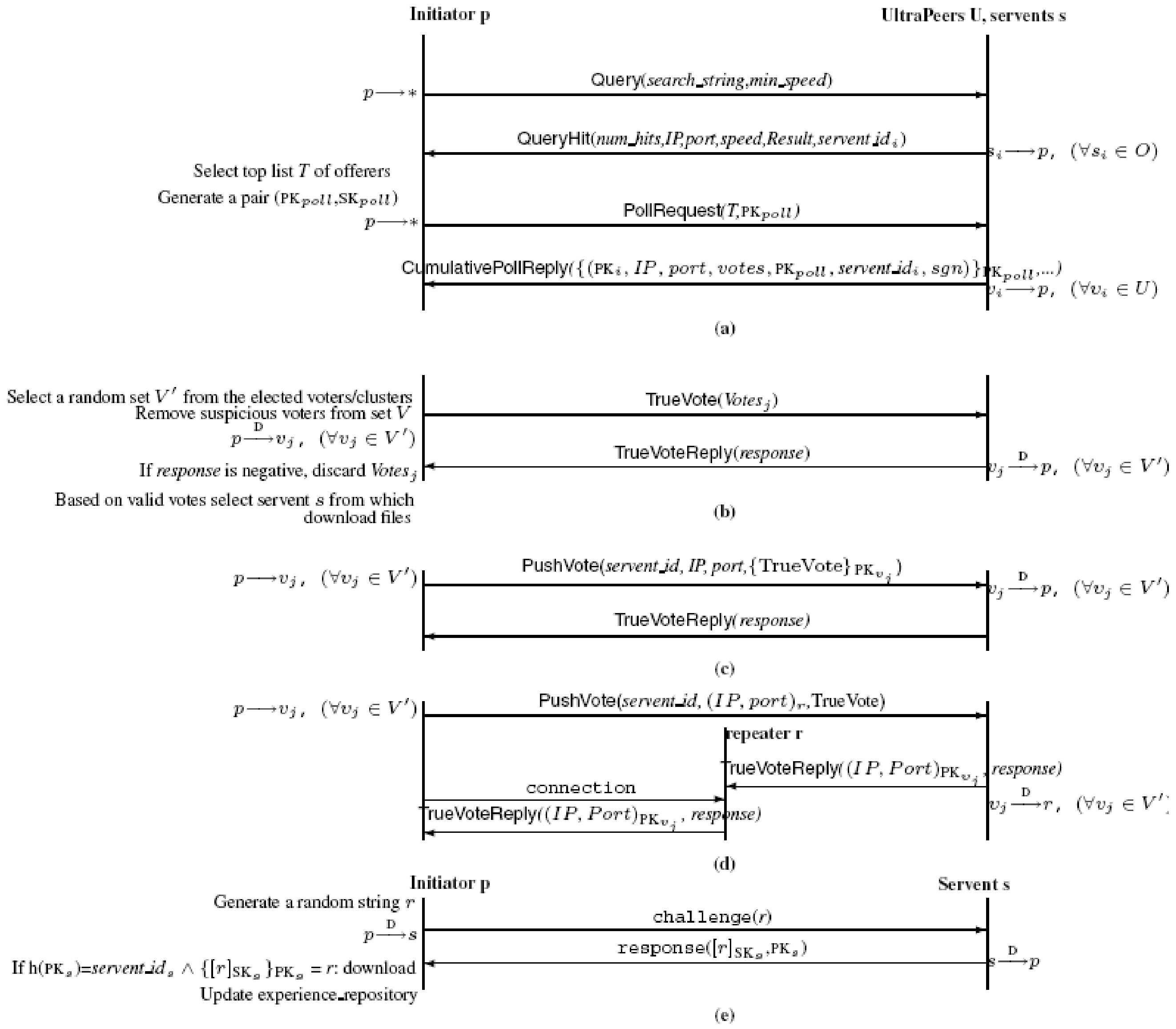

Initiator p
UltraPeers U, servents s
$p \longrightarrow *$
Query(search_string,min_speed)
QueryHit(num_hits,IP,port,speed,Result,servent_id$_i$)
$s_i \longrightarrow p,\ (\forall s_i \in O)$
Select top list T of offerers
Generate a pair (PK$_{poll}$,SK$_{poll}$)
$p \longrightarrow *$
PollRequest(T,PK$_{poll}$)
CumulativePollReply($\{(PK_i, IP, port, votes, PK_{poll}, servent\_id_i, sgn)\}_{PK_{poll}}$,...)
$v_i \longrightarrow p,\ (\forall v_i \in U)$
(a)
Select a random set V' from the elected voters/clusters
Remove suspicious voters from set V
$p \xrightarrow{D} v_j,\ (\forall v_j \in V')$
TrueVote(Votes$_j$)
TrueVoteReply(response)
If response is negative, discard Votes$_j$
$v_j \xrightarrow{D} p,\ (\forall v_j \in V')$
Based on valid votes select servent s from which download files
(b)
$p \longrightarrow v_j,\ (\forall v_j \in V')$
PushVote(servent_id, IP, port,$\{TrueVote\}_{PK_{v_j}}$)
$v_j \xrightarrow{D} p,\ (\forall v_j \in V')$
TrueVoteReply(response)
(c)
$p \longrightarrow v_j,\ (\forall v_j \in V')$
PushVote(servent_id, $(IP, port)_r$,TrueVote)
repeater r
TrueVoteReply($(IP, Port)_{PK_{v_j}}$, response)
connection
$v_j \xrightarrow{D} r,\ (\forall v_j \in V')$
TrueVoteReply($(IP, Port)_{PK_{v_j}}$, response)
(d)
Initiator p
Servent s
Generate a random string r
$p \xrightarrow{D} s$
challenge(r)
response($[r]_{SK_s}$,PK$_s$)
$s \xrightarrow{D} p$
If h(PK$_s$)=servent_id$_s$ $\wedge$ $\{[r]_{SK_s}\}_{PK_s} = r$: download
Update experience_repository
(e)

Figure 1. SupRep protocol: query and poll (a), vote verification (b)-(d), and resource download (e)

Some Lessons from the Past

- Always think about the possibility of DNS Poisoning in Caches (Refer ***Using** the **Domain Name System** for **System** Break-ins - Bellovin)*
- IP Spoofing Attacks
- DoS Attacks
- Some other Ideas ex: using the information for the compromised machines and servers (in Zombie Zones)

Spam Free , Phish Free, Reputed Safe Net?

Bad Reputation