



Spam filtering

Peter Likarish

Based on slides by EJ Jung

11/03/10

What is spam?



◆ An unsolicited email

- equivalent to Direct Mail in postal service
- UCE (unsolicited commercial email)
- UBE (unsolicited bulk email)
- 82% of US email in 2004 [MessageLabs 2004]

◆ Etymology

- Monty Python's spam episode
- "ham" = not spam



Historical events

◆ May 3rd, 1978

- Sent to Arpanet west coast users
- Invitation to see new models of DEC-20
- Violated ARPA's user policy
- Vendors were chastised & ceased (for then)

◆ March 31st, 1993 – first UUCP spam

- Unix-to-Unix Copy

◆ April, 1994 – Green Card Lottery spam

- Message posted to EVERY UUCP group



Who's the victim?

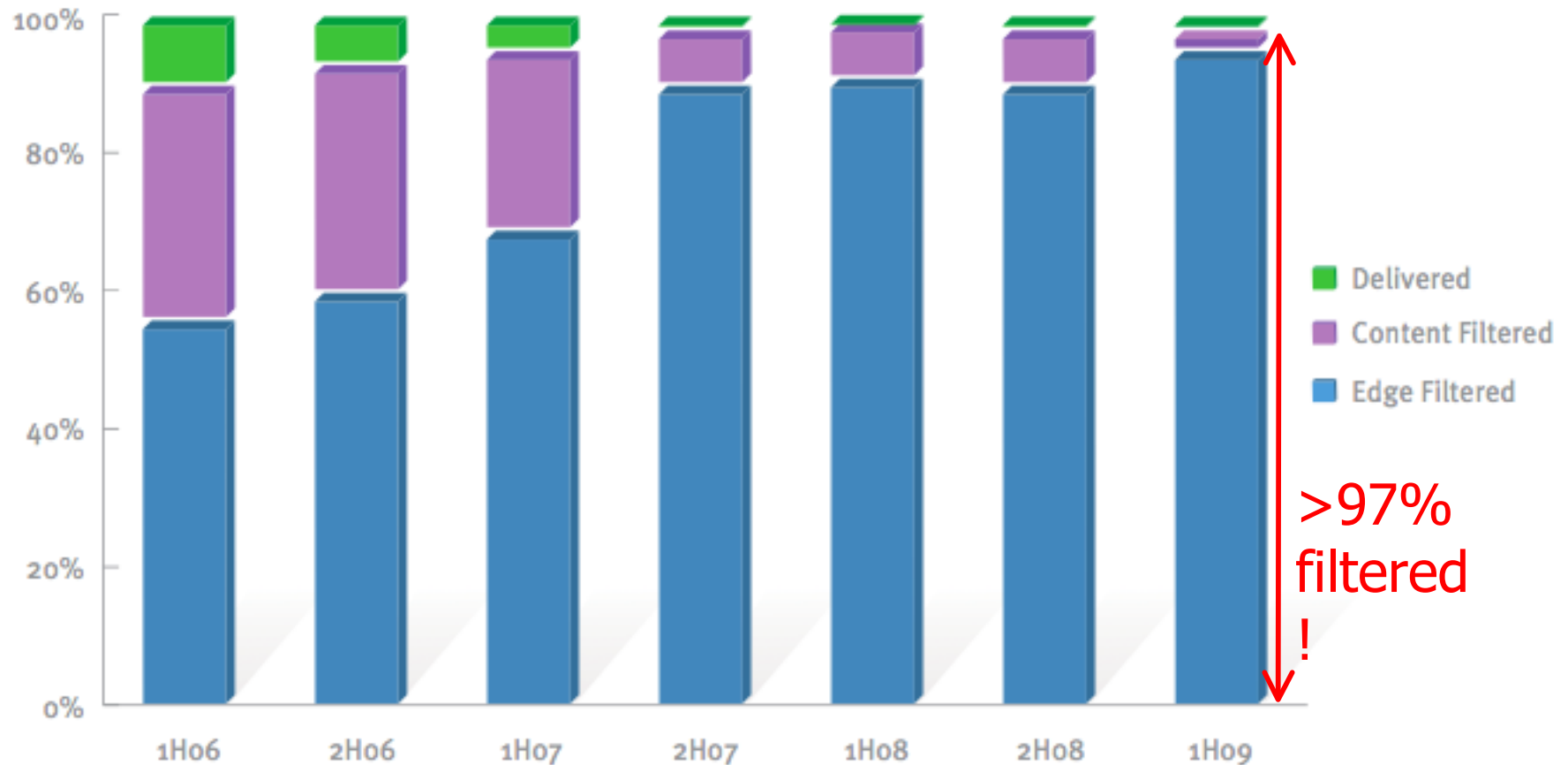
- ◆ Internet Service Provider (ISP)
 - much higher bandwidth usage
- ◆ Email service providers
 - need to support larger disk space
 - need to deploy anti-spam measures
- ◆ Businesses
 - time spent on spam
 - damage by spams and viruses and worms
- ◆ Us?



Does spam bother you?

- ◆ Potentially malicious
 - viruses, worms
 - phishing emails
- ◆ Fill up the disk space
- ◆ Does it bother you more than DMs?
 - if so, why?

How much email is spam?



Source: Microsoft Security Intelligence Report '09

(<http://www.microsoft.com/downloads/details.aspx?FamilyID=037f3771-330e-4457-a52c-5b085dc0a4cd&disclavlang=en>)



Other numbers

◆ Spamhaus

- >90%
- http://www.spamhaus.org/effective_filtering.html

◆ Postini (Google owned)

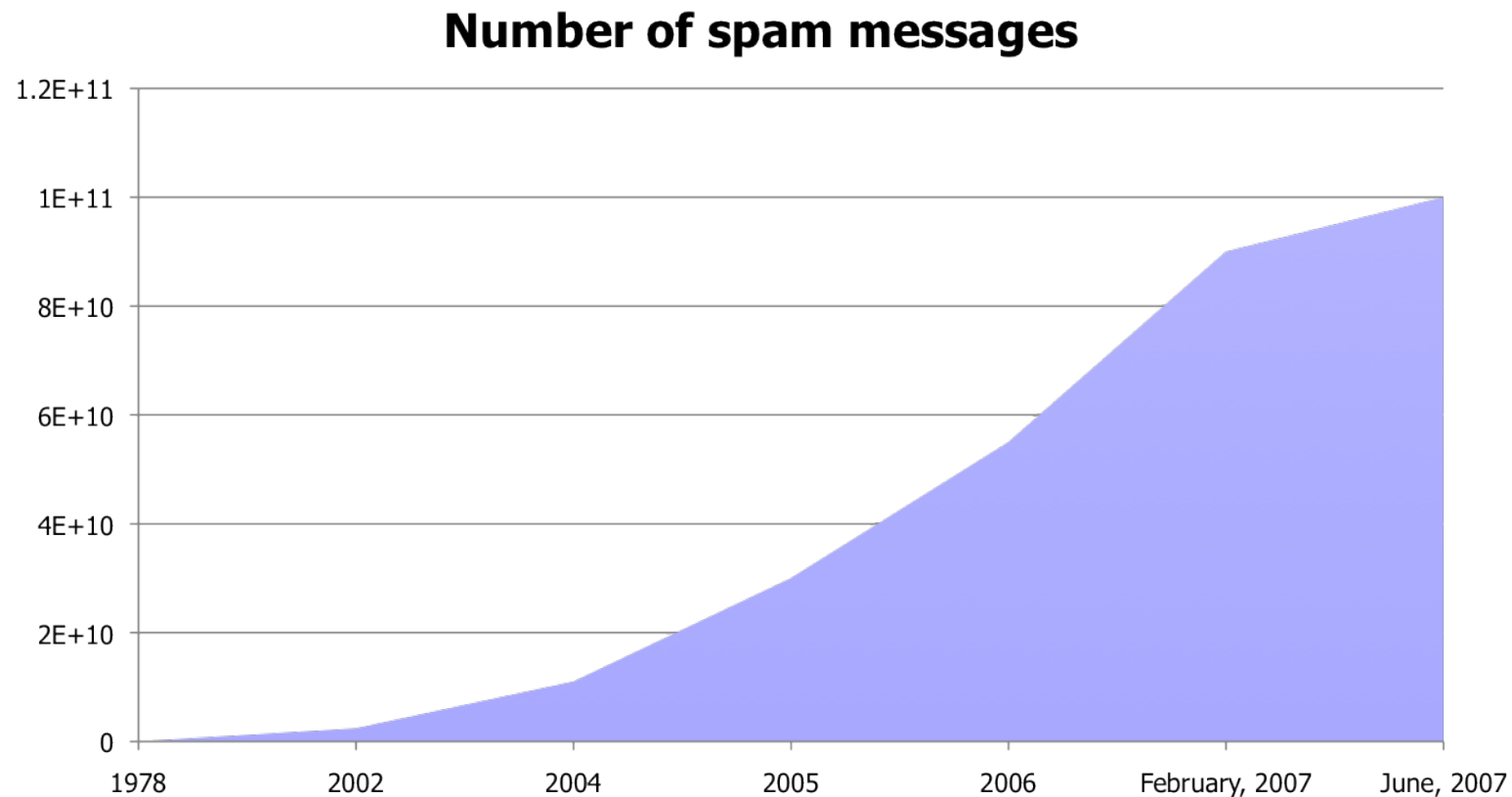
- >90%-95%
- <http://googleenterprise.blogspot.com/search/label/spam%20and%20security%20trends>

◆ Consensus: >90% of all email is spam



How many messages?

◆ Is (was) exponentially increasing





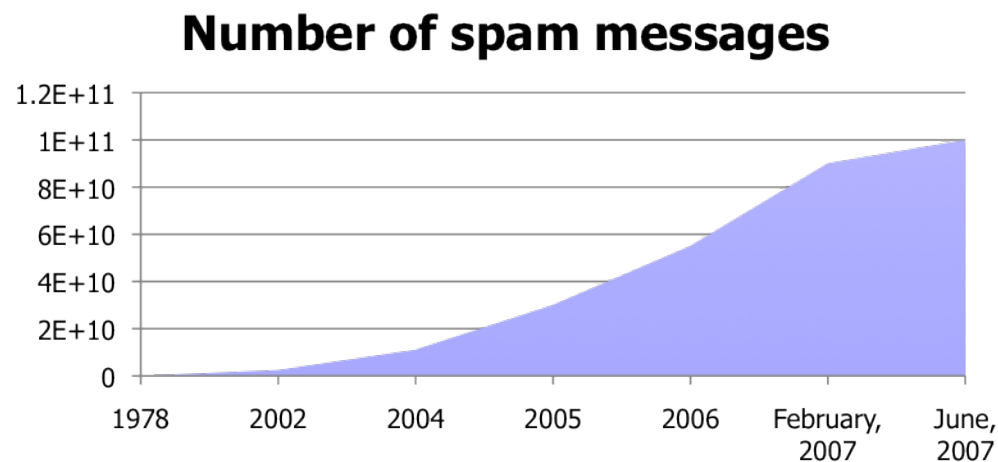
Cost of spam

- ◆ *Very* back of the envelope...
- ◆ 100,000,000,000 messages/month
- ◆ Recipient
 - \$0.025
 - \$2.5 billion/month for recipients
- ◆ Sender
 - \$0.00001 for sender to generate
 - \$1 million/month for sender
- ◆ Profit?
 - 1:200,000 = 500,000 sales. \$2 is break-even point



Legal actions

- ◆ CAN-SPAM Act of 2003
 - UBE must be labeled
 - Must have opt-out option
- ◆ Some people went to jail...
- ◆ Some ISPs have been shut down





Some ways to blocking spam

◆ Network-level

- DNS blackholes/blacklisting
- Edge filtering

◆ Content-level

- Rule-based
- checksum
- Machine learning
 - Probabilistic
 - SVM

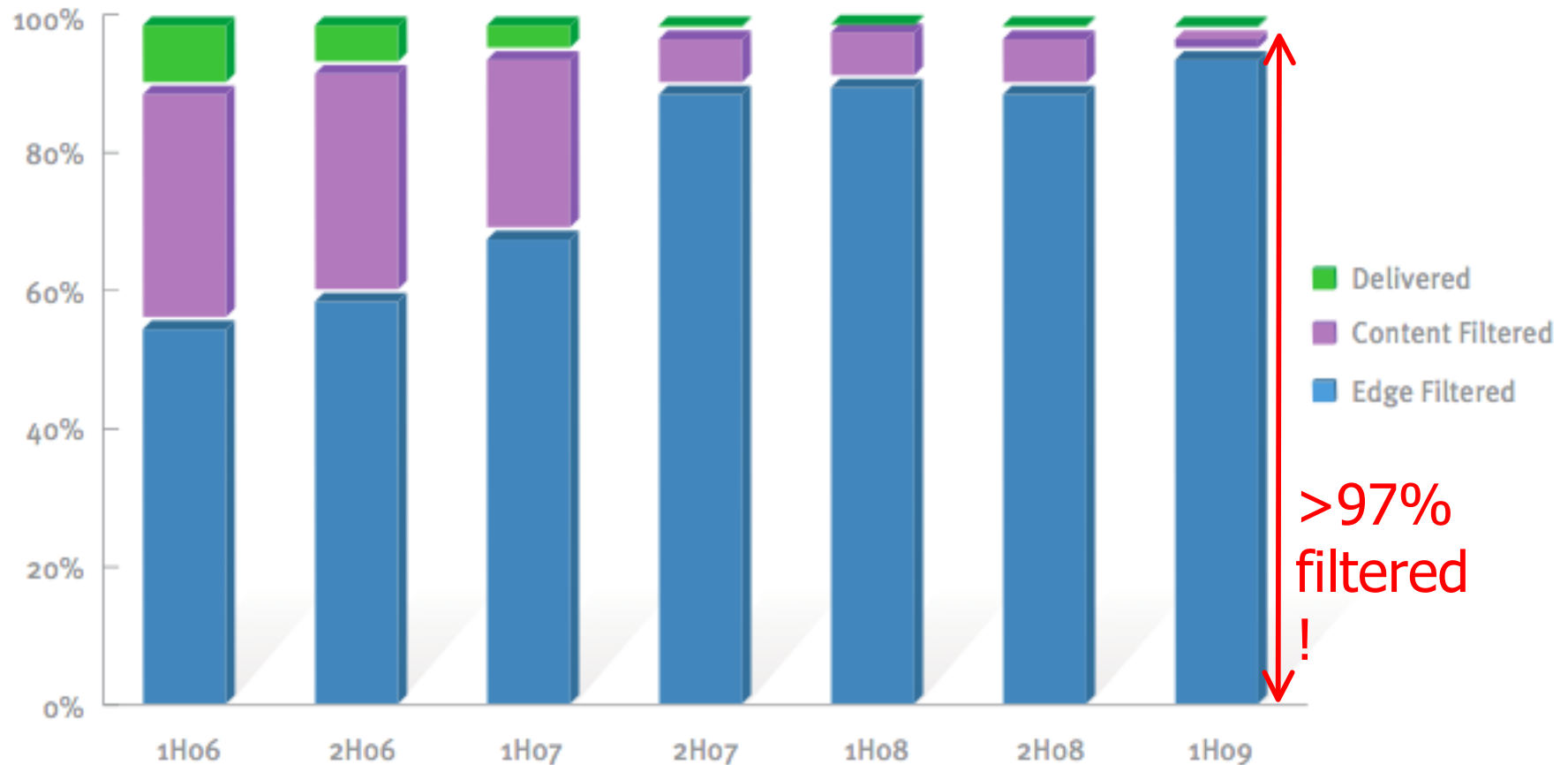
◆ Cost-based approach



DNS filtering

- ◆ Blacklist = list of known spammers
 - IP addresses/domains/senders
 - Top five spamming ISPs account for > 80% of spam traffic!
- ◆ Whitelist = list of known good senders
 - IP addresses/domains/senders
- ◆ Further reading:
 - Zhang et al. Highly Predictive Blacklisting
 - (<http://www.cyber-ta.org/releases/HPB/HighlyPredictiveBlacklists-SRI-TR-Format.pdf>)

How much email is spam?

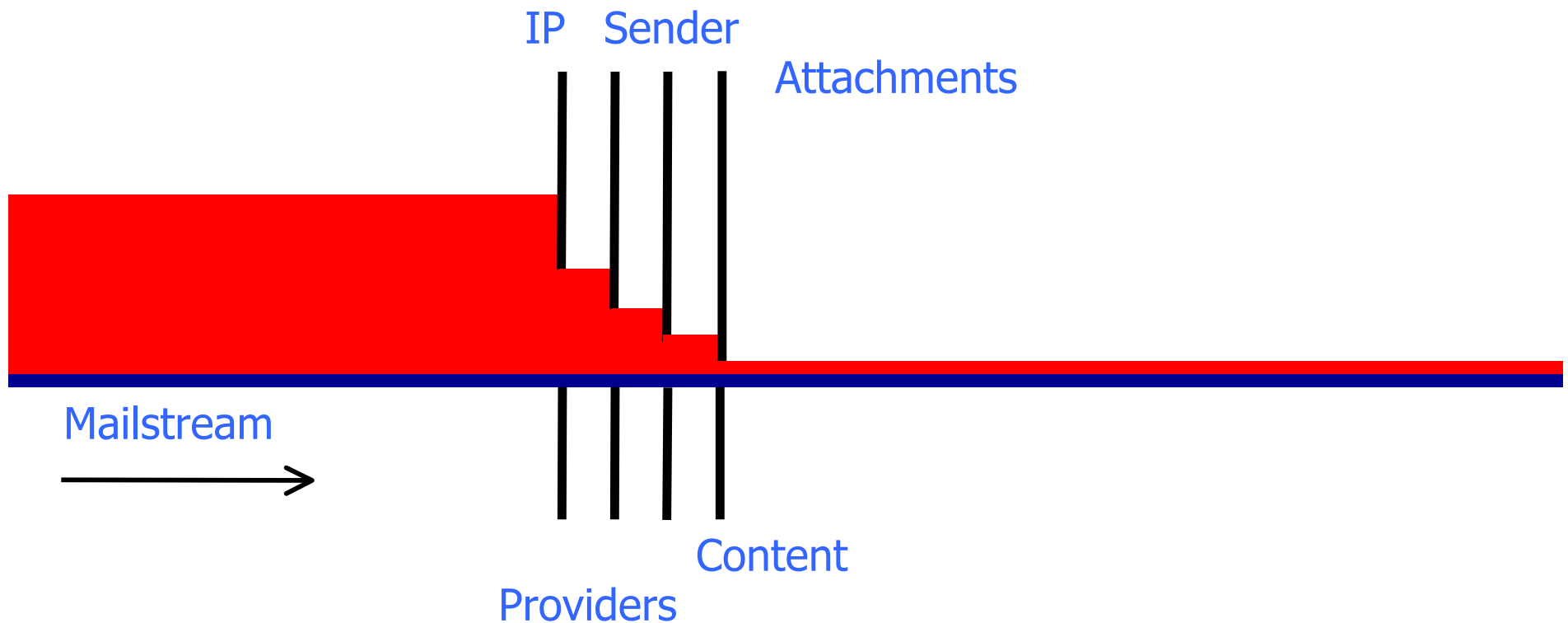


Source: Microsoft Security Intelligence Report '09

(<http://www.microsoft.com/downloads/details.aspx?FamilyID=037f3771-330e-4457-a52c-5b085dc0a4cd&disclavlang=en>)

Edge filtering

- ◆ Filtering at “edges” of network.
 - Ingress: filtering traffic entering your network
 - Egress: filtering traffic exiting your network





Content filtering

◆ Rule-based

- regular expressions on headers, contents, ...
- blacklist
- Hash/checksum database

◆ Bayesian probability/networks

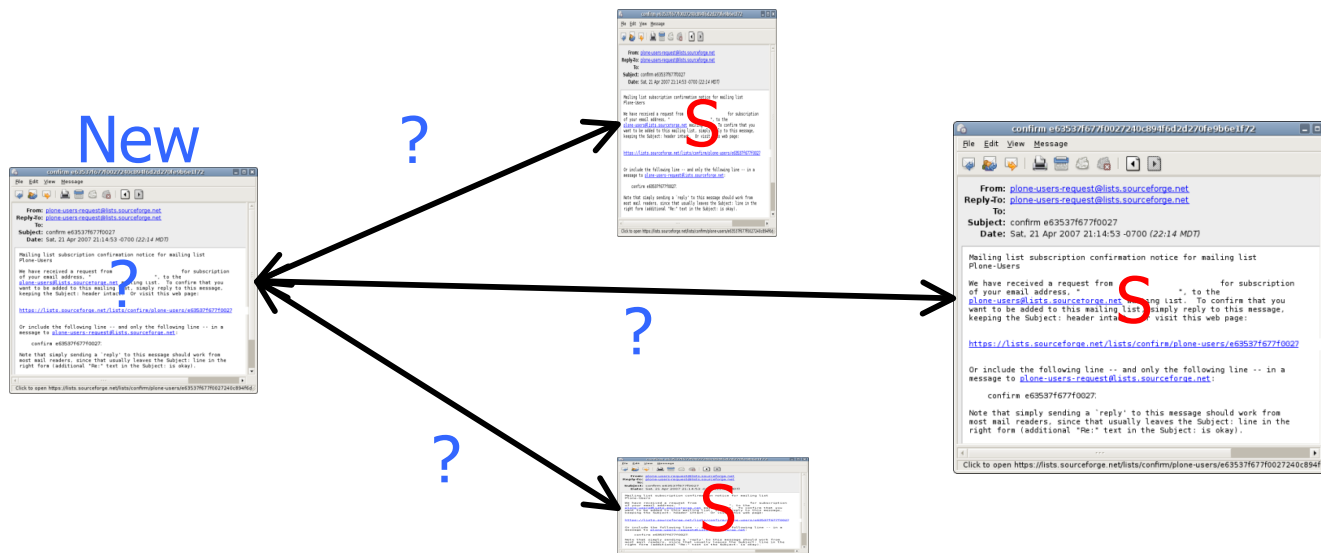
- probabilistic approach

◆ Support Vector Machine

- High dimension hyper-plane!

◆ Given new email message do I deliver or trash?

Spam I've seen





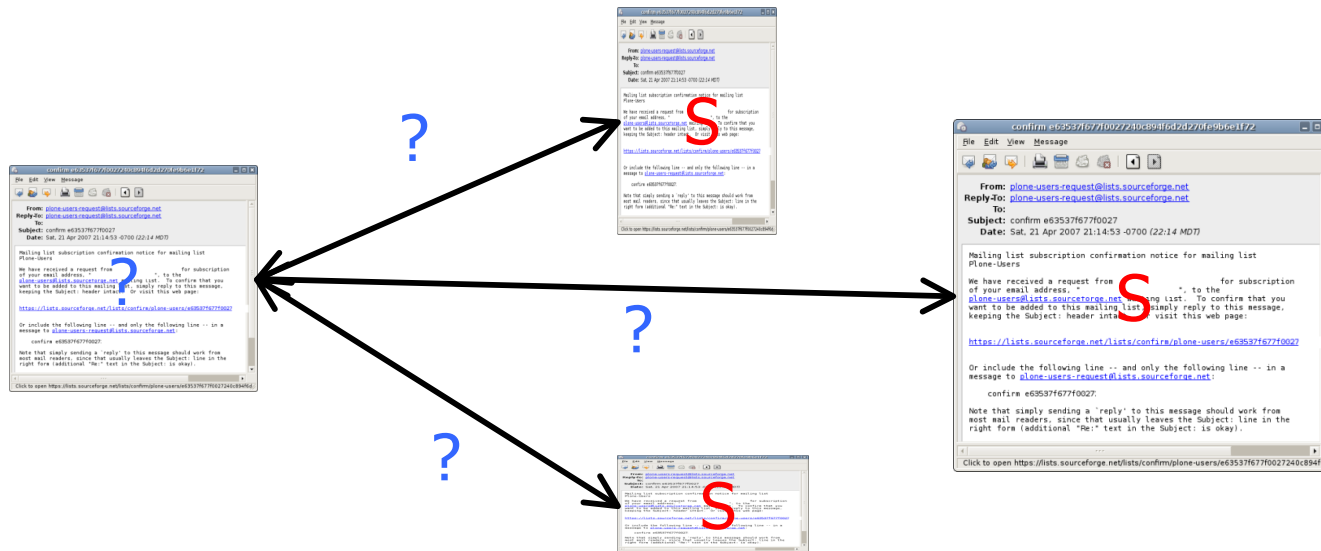
Rule-based filtering

- ◆ Rules are expressed as regular expressions (or SNORT traces)
 - If contains "refinancing" & "mortgage", trash
 - If contains non-English alphabets, trash
 - If attachment type = executable, trash

- ◆ Limitations
 - Difficult to write rules general enough to catch spam but not legit email
 - often miss obvious tricks
 - misspelling on purpose

Checksums – Fuzzy hashing

- ◆ Goal: resilient to spelling changes
- ◆ Does it look like spam I've already seen?





Fuzzy vs. non-fuzzy hashing

◆ Traditional hashing

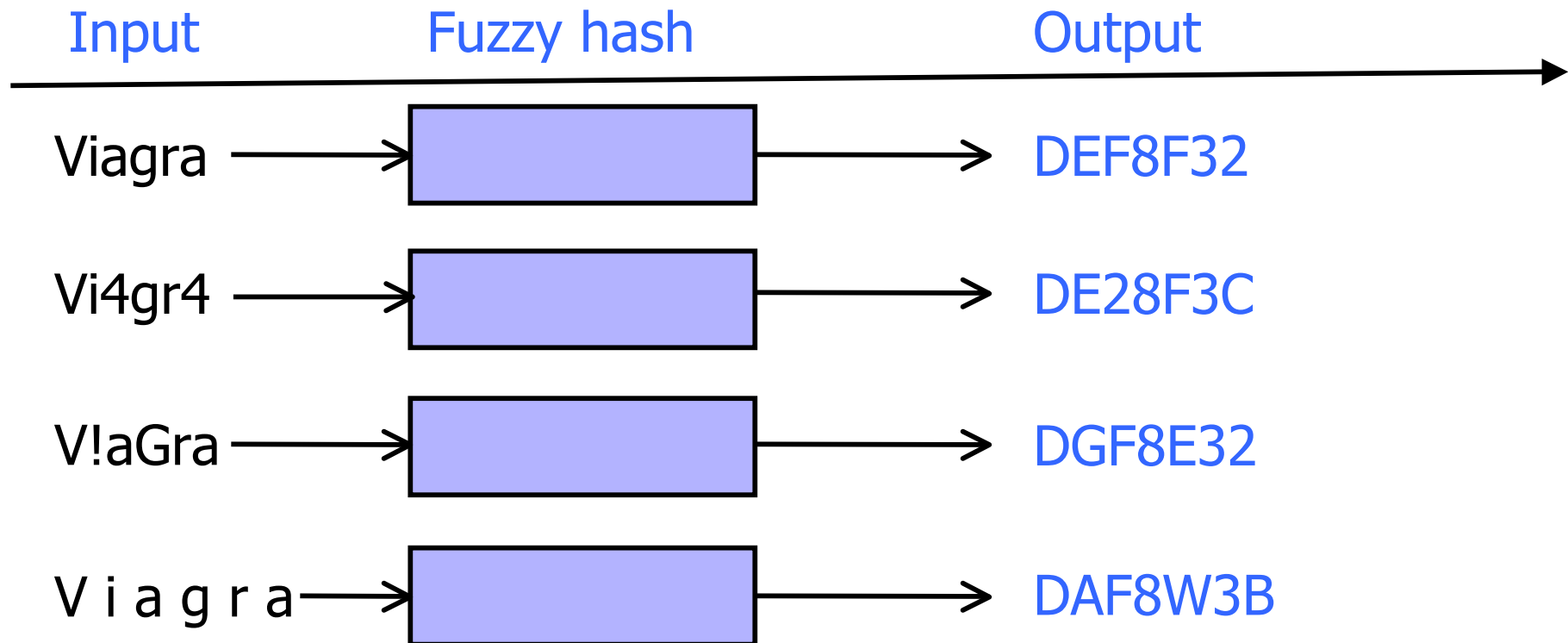
- Same: Fixed length output
- Different: Unique input = (nearly) unique output
- 1 bit change in input = COMPLETELY different hash

◆ Fuzzy hashing

- Same: Fixed length output
- Different: "mostly the same" = SAME value



Fuzzy hashing example



Output = fixed length. Look at distance between outputs.
Shorter distance = more similar. Input = message



Bayesian Theorem

- ◆ Conditional probability

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

- ◆ Bayes' Theorem

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

Bayesian in Spam [Graham 02, Sahami 98]

- ◆ Given a word “stock” in an email, what is the probability of this email being spam?

we can compute these from the sample set of emails

$$P(spam | stock) = \frac{P(stock | spam)P(spam)}{P(stock)}$$

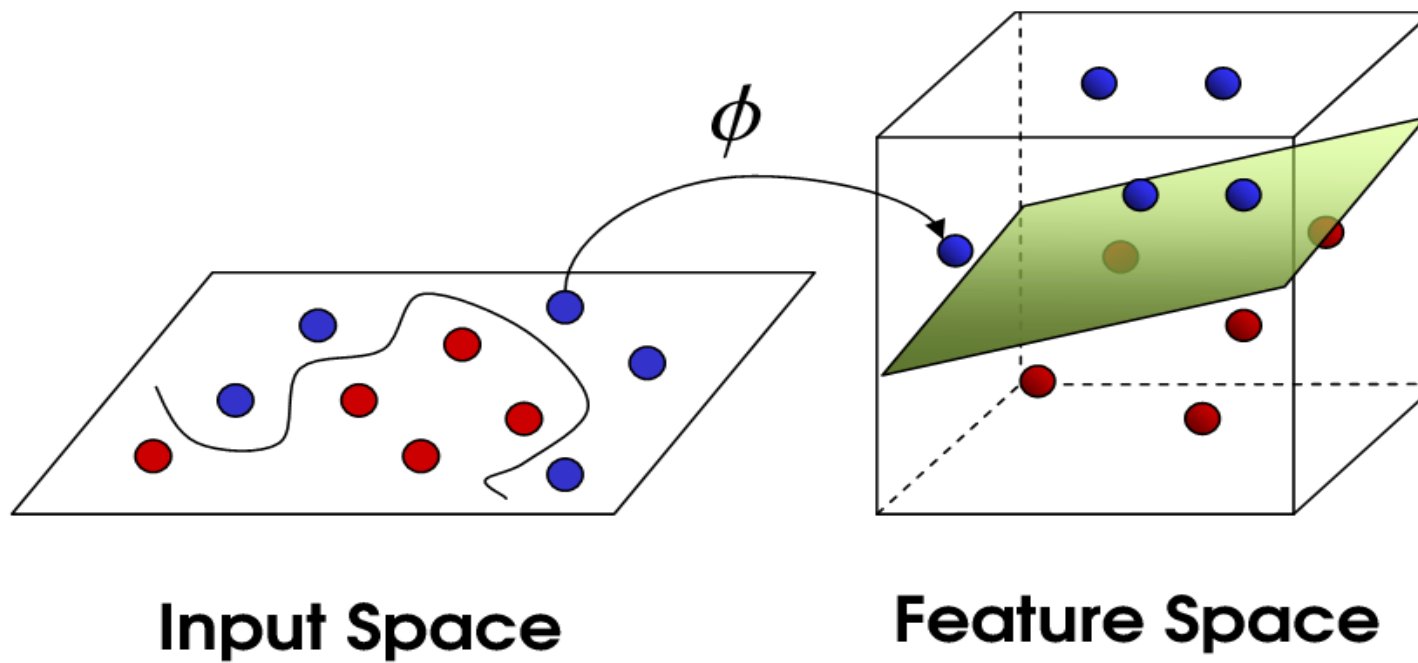
More words

- ◆ Does more words mean higher probability?
 - related words such as stock, jackpot, blue chip, ...
 - not-related words such as stock, diet, ...
 - n words require $O(2^n)$ probabilities to compute
- ◆ Naïve Bayesian filter
 - assumes the independence between words
 - $O(n)$ probabilities to compute

$$P(\text{spam} / \text{stock}, \text{jackpot}) = \frac{P(\text{stock}, \text{jackpot} \mid \text{spam})P(\text{spam})}{P(\text{stock}, \text{jackpot})}$$

Support Vector Machine

- ◆ Given records, plot them in n-dimension spaces and find a “plane” that divides the plots the best.





Feature selection

- ◆ How do we choose which tokens to examine?
- ◆ Words alone are not sufficient
 - phrases, punctuation, ...
 - sender's email address, IP address, domain name, ...
 - URLs, images, tags, ...
- ◆ More sophisticated conditions
 - attachment?
 - contains images?



Hybrid approaches

◆ SpamAssassin

- open-source
- combination of rule-based and Bayesian
- commonly run at SMTP server
- can be applied for an individual user

◆ Vipul's Razor

- Collaborative, distributed, checksum (statistical & randomized signature-based) anti-spam solution
- Commercial version: Cloudmark, Inc



Charging for email

- ◆ If you can't win the game, change the rules...
- ◆ Would you pay \$0.0025 to send an email?

- ◆ If you sent 50 emails/day:
 - ~\$46/year

- ◆ Spammers:
 - $100,000,000,000 * \$0.0025 = \$250,000,000$ dollars
 - Assuming adoption rate of 1:200,000
 - Break even = \$1,250