Portland State UNIVERSITY

MetaCAPTCHA: A Metamorphic Throttling Service for the Web

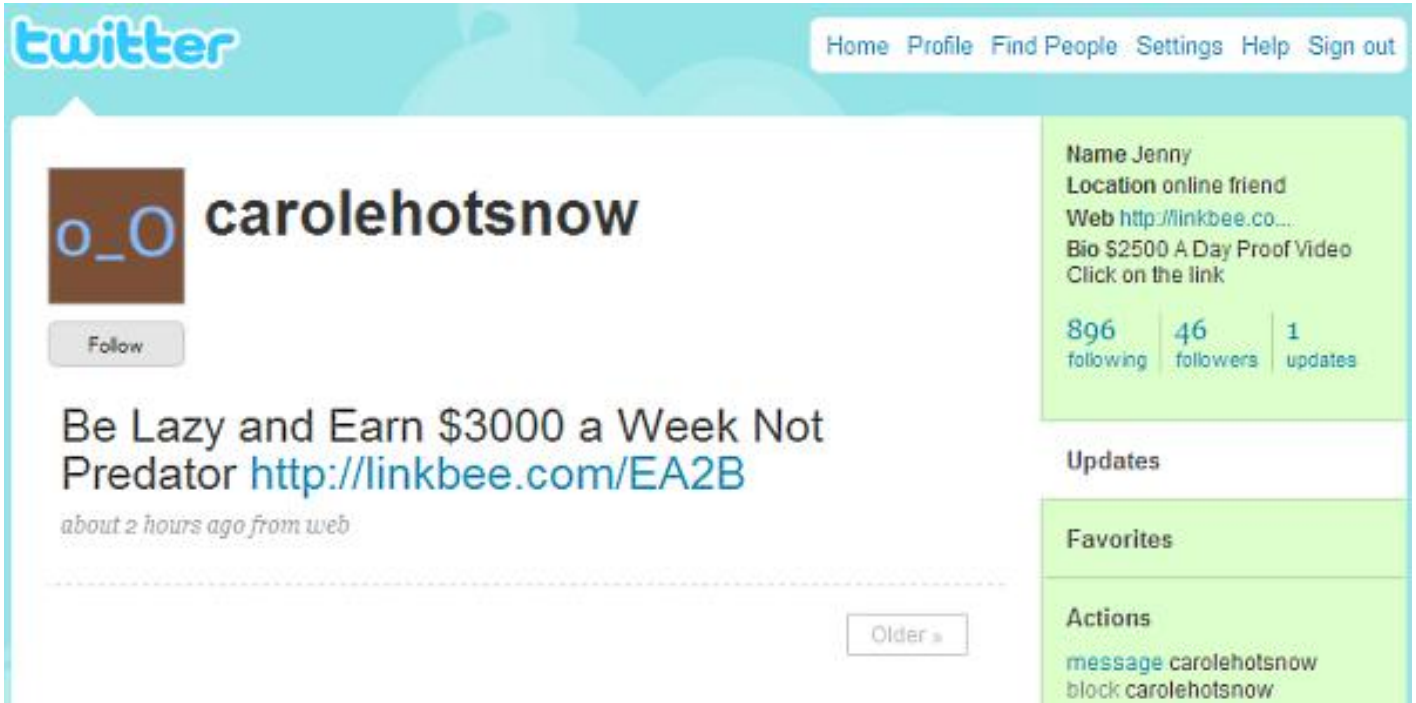
Akshay Dua, Thai Bui, Tien Le, Nhan Huynh, Wu-chang Feng
{akshay, buithai, letien, nhhuyng, wuchang}@cs.pdx.edu

Portland State University

October 21, 2014

Traditional email spam

The market is moving to social Spam



Difficult to detect kind of spam

As a result

Email spam is *reducing* but social spam is *edging up*, with

As a result

Email spam is *reducing* but social spam is *edging up*, with

Large volume Four million Facebook users receive spam each day in 2011 [5, 9]

As a result

Email spam is *reducing* but social spam is *edging up*, with

Large volume Four million Facebook users receive spam each day in 2011 [5, 9]

Fast growth Cost businesses $20.5 billion annually and projected to $198 billion in the next four years [12]

As a result

Email spam is *reducing* but social spam is *edging up*, with

Large volume Four million Facebook users receive spam each day in 2011 [5, 9]

Fast growth Cost businesses $20.5 billion annually and projected to $198 billion in the next four years [12]

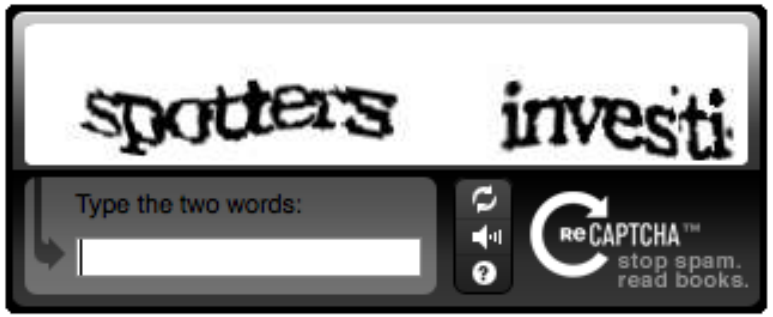
High conversion rate The "clickthrough" rate of spam URLs on Twitter was almost two times higher than email spam in 2010 [6]

Two main methods

CAPTCHAs

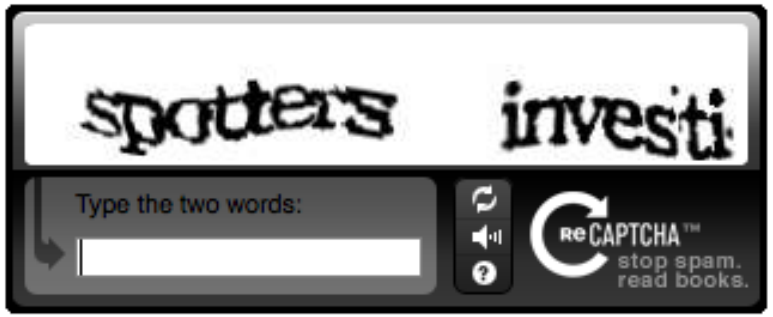
proof-of-work

CAPTCHAs



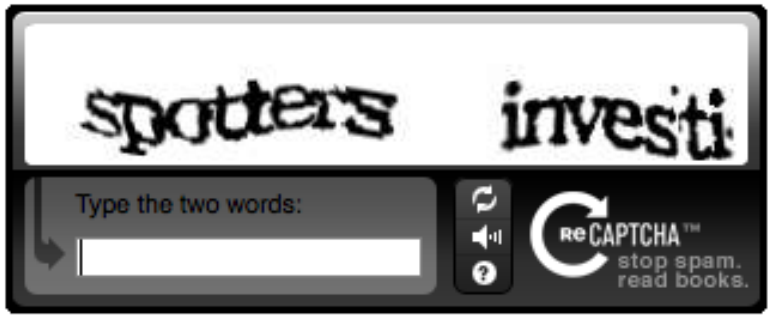
- can prevent bots effectively ..

CAPTCHAs



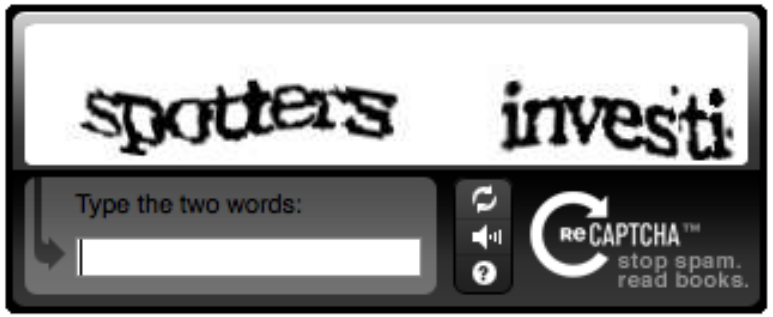
- can prevent bots effectively .. as long as there aren't OCR algorithms that can solve it [13]

CAPTCHAs



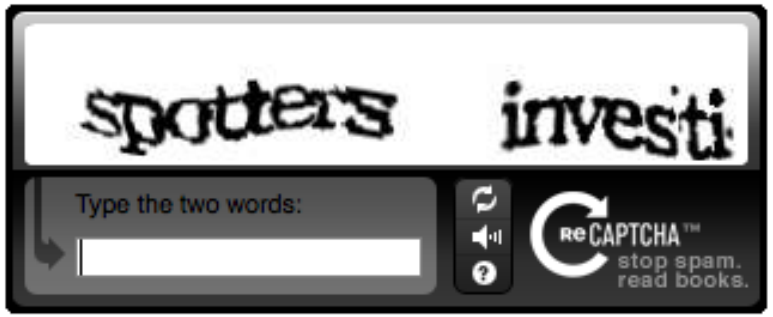
- can prevent bots effectively .. as long as there aren't OCR algorithms that can solve it [13]
- no way to have variable cost of solving [10, 11]

CAPTCHAs



- can prevent bots effectively .. as long as there aren't OCR algorithms that can solve it [13]
- no way to have variable cost of solving [10, 11]
- can only use for infrequent transactions due to the usability burden [14]

CAPTCHAs



- can prevent bots effectively .. as long as there aren't OCR algorithms that can solve it [13]
- no way to have variable cost of solving [10, 11]
- can only use for infrequent transactions due to the usability burden [14]

= annoying

Proof-of-work

Proof-of-work

- does not have CAPTCHA's usability issues

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions
- thus, can have variable cost of solving

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions
- thus, can have variable cost of solving

= nice right?

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions
- thus, can have variable cost of solving

= nice right? but

- many proposed systems do not have an accurate user reputation

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions
- thus, can have variable cost of solving

= nice right? but

- many proposed systems do not have an accurate user reputation
- or, are too tightly integrated with a given application [3]

Proof-of-work

- does not have CAPTCHA's usability issues
- can be used in frequent transactions
- thus, can have variable cost of solving

= nice right? but

- many proposed systems do not have an accurate user reputation
- or, are too tightly integrated with a given application [3]

= boo :(

"License" and "tax" spam

Combines the strength of CAPTCHA and proof-of-work as puzzles

"License" and "tax" spam

Combines the strength of CAPTCHA and proof-of-work as puzzles

Variable Cost Function The more you spam the "harder" puzzles you have to solve. Uses a Bayesian reputation system

"License" and "tax" spam

Combines the strength of CAPTCHA and proof-of-work as puzzles

Variable Cost Function The more you spam the "harder" puzzles you have to solve. Uses a Bayesian reputation system

Secure The solver code is metamorphic: changing code randomly in each transaction

"License" and "tax" spam

Combines the strength of CAPTCHA and proof-of-work as puzzles

Variable Cost Function The more you spam the "harder" puzzles you have to solve. Uses a Bayesian reputation system

Secure The solver code is metamorphic: changing code randomly in each transaction

Easy to use Easy to install & manage allowing the addition or removal of "ineffective" puzzles

Communication Protocol

Authentication = Kerberos model

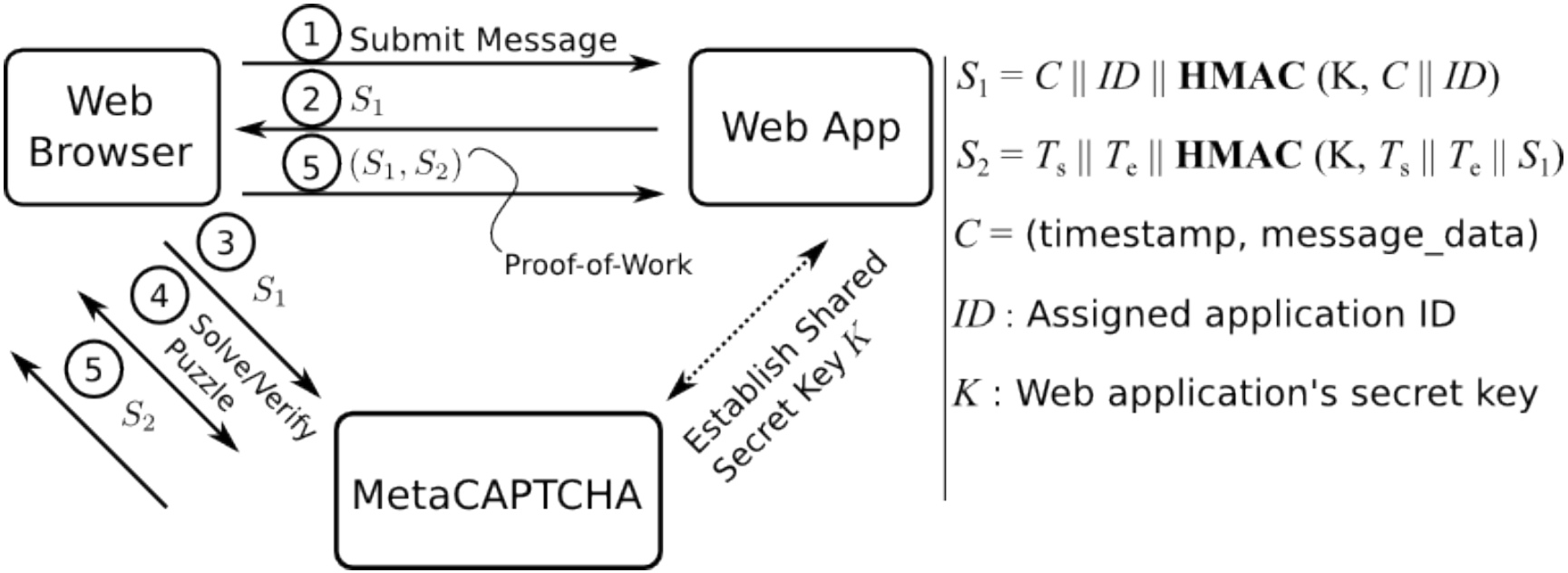


Figure 1: MetaCAPTCHA puzzle delivery and solution verification

Using Bayesian model

Training Data

Using Bayesian model

Training Data
$\rightarrow$ Naive Bayes classifier

Using Bayesian model

Training Data
$\rightarrow$ Naive Bayes classifier
$\rightarrow$ Reputation score r between 0 and 1

Using Bayesian model

Training Data
$\rightarrow$ Naive Bayes classifier
$\rightarrow$ Reputation score r between 0 and 1
$\rightarrow$ Puzzle difficult t

Using Bayesian model

Training Data
$\rightarrow$ Naive Bayes classifier
$\rightarrow$ Reputation score r between 0 and 1
$\rightarrow$ Puzzle difficult t
$\rightarrow$ Random generated puzzles

Using Bayesian model

Training Data
→ Naive Bayes classifier
→ Reputation score r between 0 and 1
→ Puzzle difficult t
→ Random generated puzzles

Client solves the puzzles until reaches the total amount of time t

Non-interactive, interactive or both

Web apps determine what puzzle types to protect their websites

Non-interactive, interactive or both

Web apps determine what puzzle types to protect their websites

Puzzle types

- Non-interactive puzzles
  - Targeted Hash-Reversal [4]
  - Modifed Time-Lock [3]

Non-interactive, interactive or both

Web apps determine what puzzle types to protect their websites

Puzzle types

- Non-interactive puzzles
  - Targeted Hash-Reversal [4]
  - Modifed Time-Lock [3]
- Interactive puzzles
  - reCAPTCHA

Non-interactive, interactive or both

Web apps determine what puzzle types to protect their websites

Puzzle types

- Non-interactive puzzles
  - Targeted Hash-Reversal [4]
  - Modifed Time-Lock [3]
- Interactive puzzles
  - reCAPTCHA
- Hybrid
  - CAPTCHA+: reCAPTCHA and Modifed Time-Lock

More puzzle types can be added/removed with no changes to the web application

Environment & Dataset

- Deployed MetaCAPTCHA on a live discussion forum active for about two months in 2012
- Had $\approx$ 2000 messages, $\approx$ 500 users, $\approx$ 100 sub-forums with $\approx$ 1000 threads

F-measure of different features

10-fold cross-validation to train on ≈ 1500 messages and test the classifier on multiple features

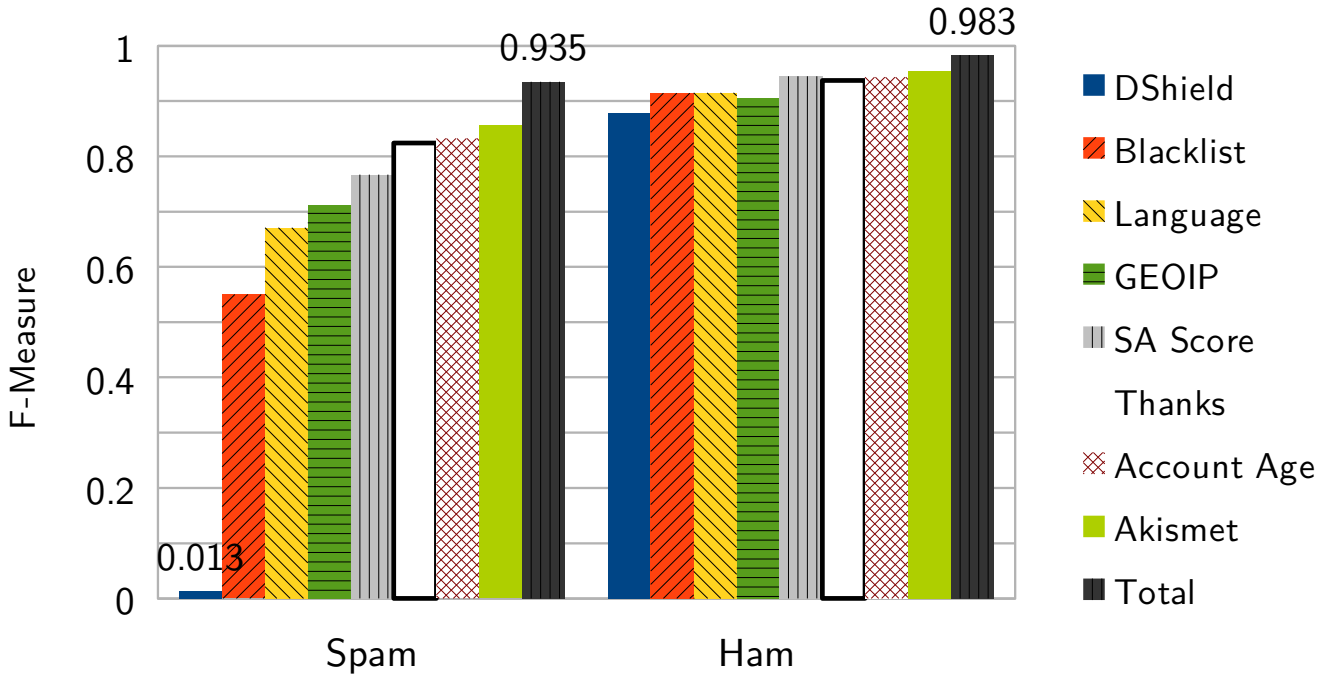


Figure 2: Using multiple features is better than using one or a few

Puzzle Difficulty Accuracy

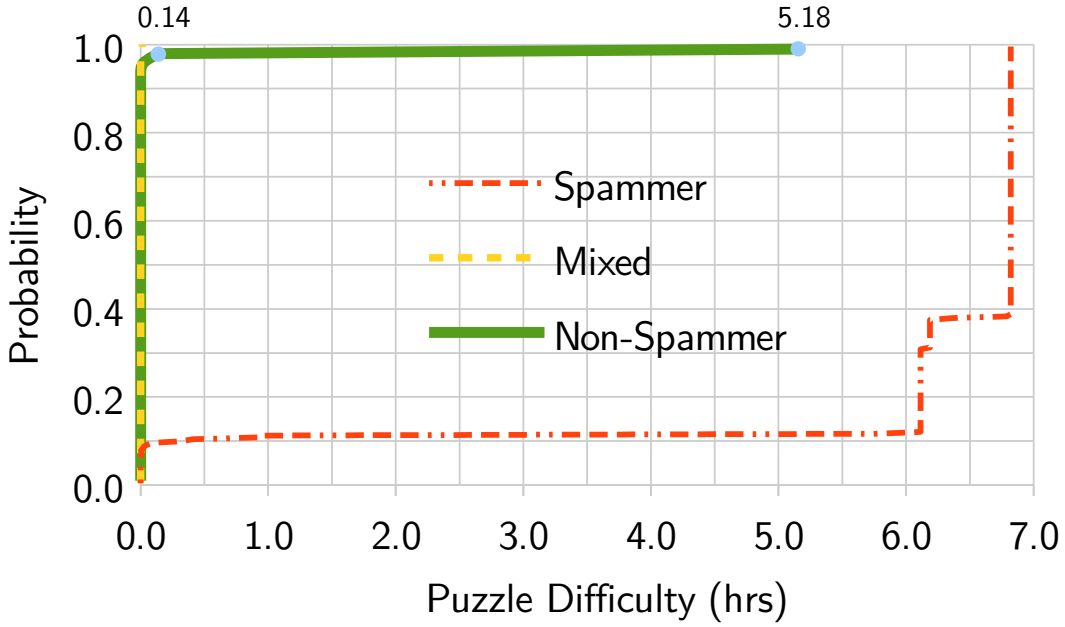


Figure 3: CDF solving time of spammers, non-spammers and mixed users

- $\approx$ 90% of spammers solved a puzzle over 6 hrs long
- $\approx$ 95% of non-spammers solved *no puzzles at all* and $\approx$ 5% spent between 7.2 secs to 8.4 minutes

Mixed users posted more ham than spam

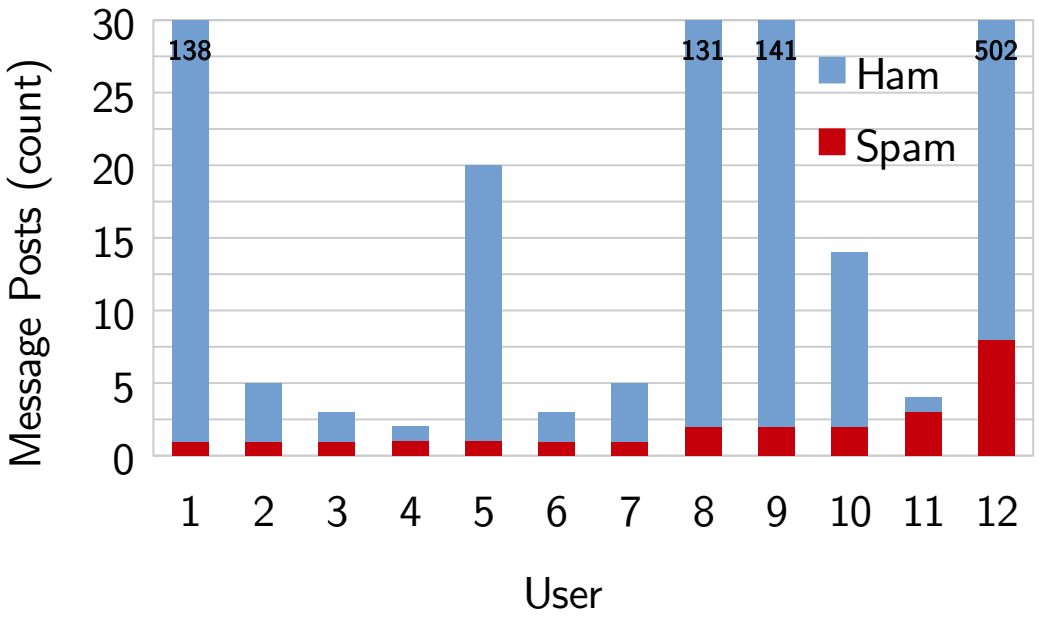


Figure 4: Distribution of spam and ham sent by mixed users. Mixed users sent very little spam (between 1 and 8) when compared to the total messages they posted. X-axis inidicates User ID

MetaCAPTCHA can really hurt spammers

- Slow down 90% of "spammers" significantly so they don't spam others
- No impact on 95% of honest users

MetaCAPTCHA can really hurt spammers

- Slow down 90% of "spammers" significantly so they don't spam others
- No impact on 95% of honest users
- Some improvement on those "unlucky users" who blamed their computers were too slow

MetaCAPTCHA can really hurt spammers

- Slow down 90% of "spammers" significantly so they don't spam others
- No impact on 95% of honest users
- Some improvement on those "unlucky users" who blamed their computers were too slow

Future Works

- Using spammer's computing resources for volunteer computing (e.g. SETI@Home)
- Bitcoin as proof-of-work; turning spammers into miners

Acknowledgment

Portland State UNIVERSITY

This work as supported by the National Science Foundation under Grant Number CNS-1017034

Question?

References I

[1] A. Back et al. Hashcash-a denial of service counter-measure. *URL: http://www. hashcash. org/papers/hashcash. pdf*, 2002.

[2] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Advances in CryptologyCRYPTO92*, pages 139–147. Springer, 1993.

[3] W. Feng and E. Kaiser. kapow webmail: Effective disincentives against spam. *Proc. of 7th CEAS*, 2010.

[4] W.-c. Feng and E. Kaiser. The case for public work. In *IEEE Global Internet Symposium, 2007*, pages 43–48. IEEE, 2007.

[5] Geoffrey A. Fowler, Shayndi Raice, Amir Efrati. Facebook, Twitter battle 'social' spam. `http://www.theaustralian.com.au/business/wall-street-journal/facebook-twitter-battle-social-spam/story-fnay3ubk-1226237108998`, Jan 2012.

[6] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[7] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *Internet Computing, IEEE*, 11(6):36–45, 2007.

[8] B. Laurie and R. Clayton. Proof-of-work proves not to work. In *The Third Annual Workshop on Economics and Information Security*, 2004.

[9] Mark Risher. Social Spam and Abuse — Annual Trend Review. `http://blog.impermium.com/2012/01/13/social-spam-and-abuse-the-year-in-review/`, Jan 2012.

[10] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: Captchas–understanding captcha-solving services in an economic context. In *USENIX Security Symposium*, volume 10, 2010.

References II

[11] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. Dirty jobs: The role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX conference on Security*, pages 14–14. USENIX Association, 2011.

[12] SPAM LAWS. Spam Statistics and Facts. `http://www.spamlaws.com/spam-stats.html`, 2011.

[13] O. R. Team. List of weaknesses. `http://ocr-research.org.ua/list.html`.

[14] J. Yan and A. El Ahmad. Usability of captchas or usability issues in captcha design. In *Proceedings of the 4th symposium on Usable privacy and security*, pages 44–52. ACM, 2008.

MetaCAPTCHA's system model

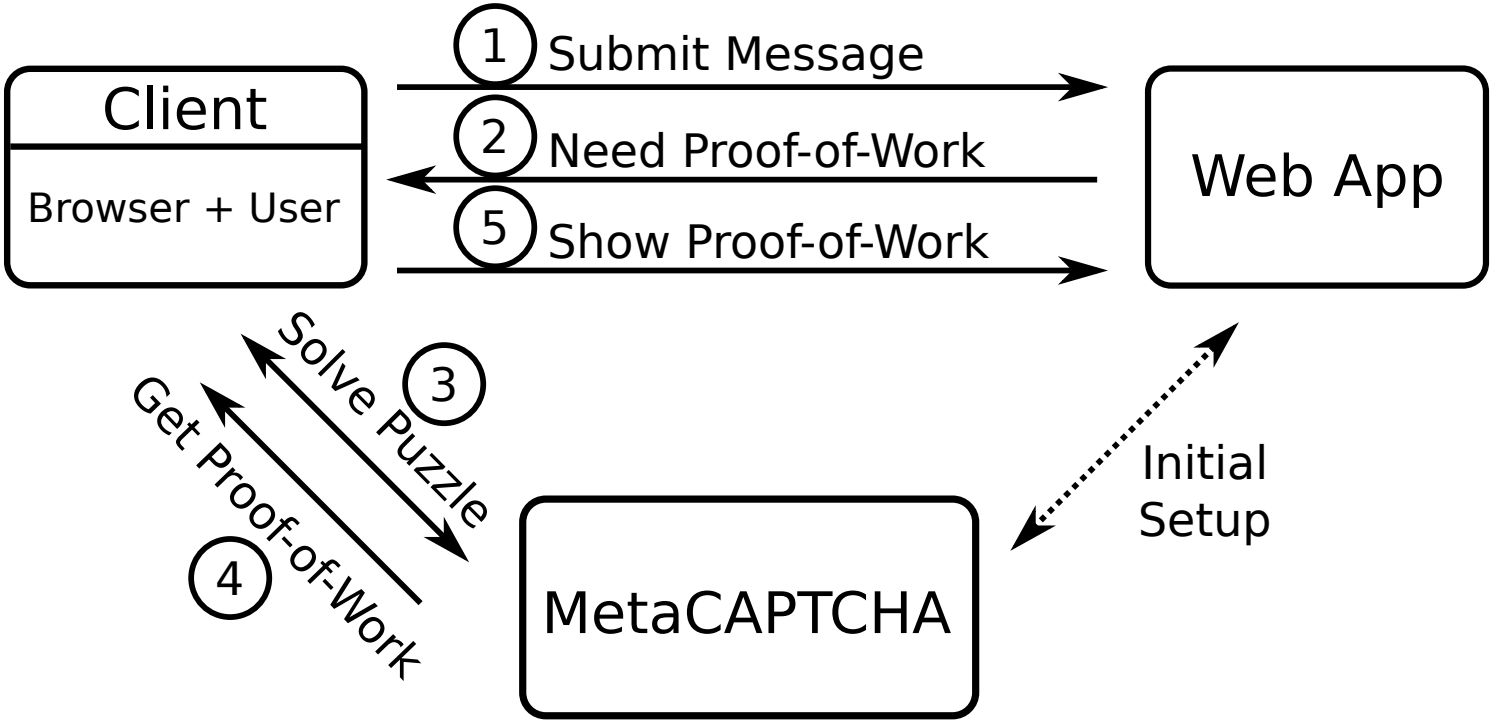


Figure 5: User's browser must show proof-of-work before the web application accepts the user's message. The dotted line indicates initial setup performed by the web application to use the MetaCAPTCHA service.

Using Bayesian model

MetaCAPTCHA calcualtes a *reputation score r* between 0 and 1

- probability that a given message is spam as determined by a Naive Bayes classifier

The *reputation score r* is translated to the *puzzle difficulty t* which is the amount of time a client must be kept busy solving puzzles [1]

$$t = (t_{max} + 1)^r - 1, t_{max} = \frac{t_p}{s_p(1-\delta)}$$

where, δ is the reduction in spam the web application is seeking (e.g. 10%) from an average amount s_p of spam messages received in time period t_p

[1] inspired by Laurie and Clayton's work on proof-of-work [8]

Solving Puzzles

Puzzles are randomly generated based on the list that is configured

- Must be solved by the user's browsers or the users
- If the solution is returned in time $t' < t$, then a new puzzle is chosen and issued
- This process is repeated until the client has computed for *at least* t amount of time

The idea behind issuing several puzzles is to ensure that no user can complete an online transaction unless they have computed for a length of time $\geq t$

Puzzles = CAPTCHAs + Proof-of-work

Proof-of-work

- First proposed by Dwork and Naor [2] to combat email spam
- Non-interactive
- Difficult to solve in terms of time & complexity, but easy to verify answers

A famous example is Hashcash [1] - a computational challenge where the computer has to find a k-bit partial hash collision on string x, given a hash function H and string y, such that the first k bits of $H(x)$ and $H(y)$ are equal

More experimental setup

- forum users divided into three categories, (i) *spammers*: (ii) *non-spammers*, and (ii) *mixed*: those who sent both spam and ham
- Here, 'users' implies the senders of messages included in ground-truth information provided by the forum.
- After the categorization, there were 99 messages sent by non-spammers, 240 messages sent by spammers, and 151 messages sent by mixed users in the test set (34% of ground-truth data picked uniformly at random).

Puzzle difficulty parameters

- $t_{max} = 6.82$ hrs based on time period $t_p = 1$ month
- number of spam messages s_p seen in that month is 1442, and a spam reduction factor $\delta = 0.6$
- $\approx 90\%$ of spammers solved a puzzle over 6 hrs long
- $\approx 5\%$ of non-spammers solved a puzzle between 7.2 secs and 8.4 minutes long
- $\approx 95\%$ of non-spammers solved *no puzzles at all*.

Solving time of spammers, non-spammers and mixed users

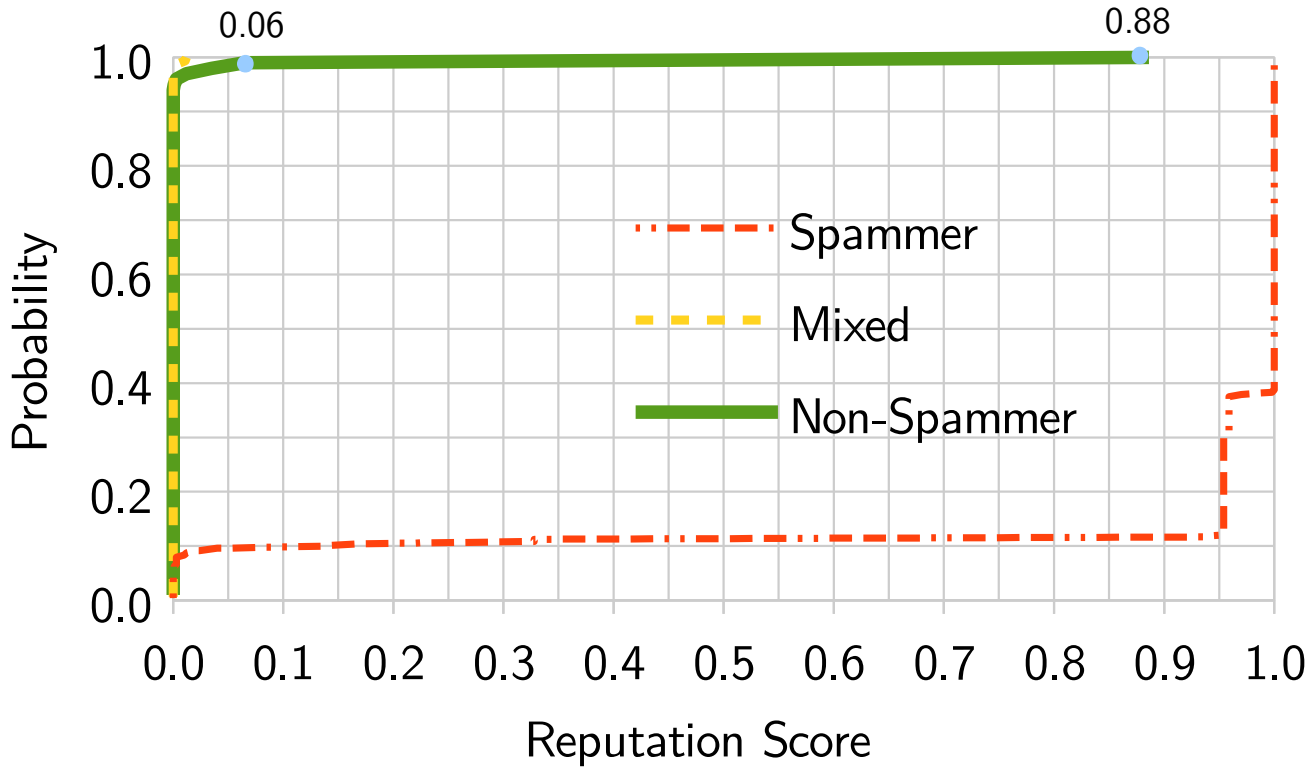


Figure 6: CDF of reputation scores assigned to spammers, non-spammers, and mixed users (those that sent at least 1 spam and 1 ham)

Reputation Score Evaluations

- $\approx$ 90% of *spammers* have reputation scores over 0.95. $\approx$ 99% of non-spammers got a reputation of 0.065 or less.
- Only one honest user suffered the ill fate of being assigned a reputation of 0.88, whereas 94% were assigned a reputation of zero — implying that they did not solve a puzzle at all!

Majority of mixed users posted more ham than spam

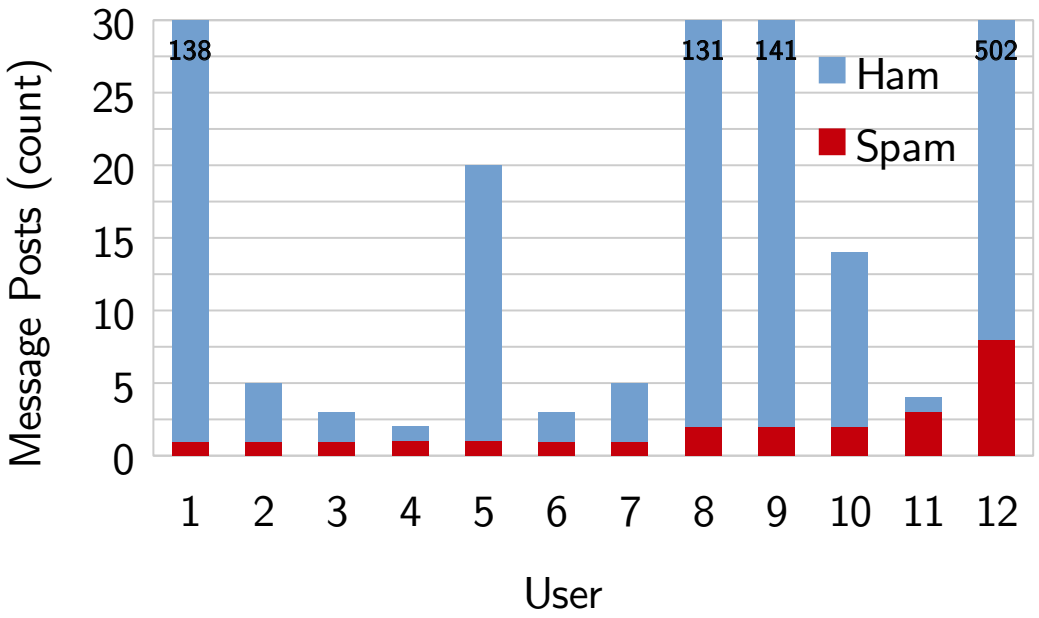


Figure 7: Distribution of spam and ham sent by mixed users. Mixed users sent very little spam (between 1 and 8) when compared to the total messages they posted. X-axis inidicates User ID