Spam Protection by using **S**ender **A**ddress **V**erification **E**xtension (SAVE)

Michael Conrad, Hans-Joachim Hof
[conrad|hof]@tm.uka.de

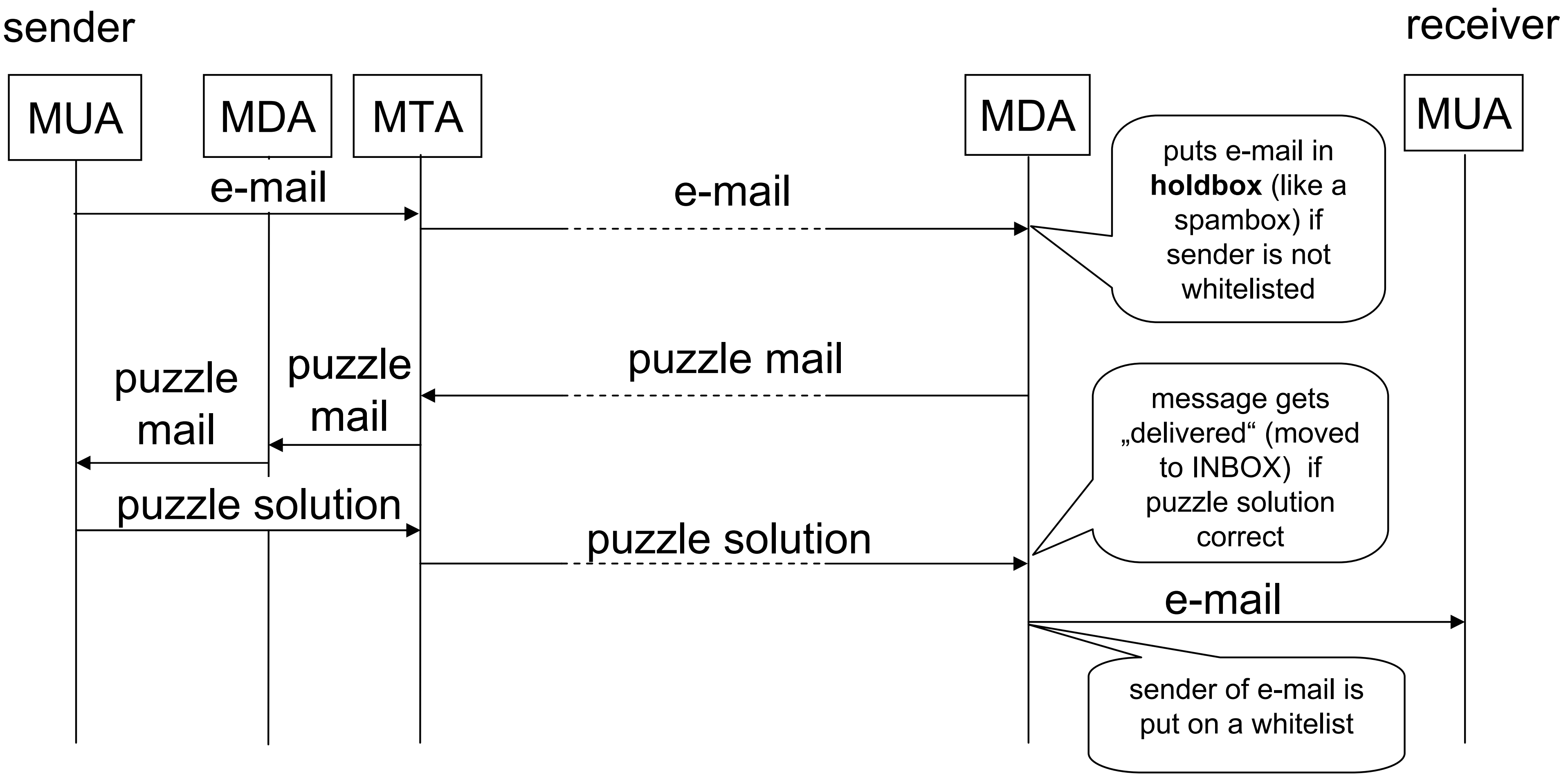
Roland Bless
bless@tm.uka.de

Institute of Telematics, Universität Karlsruhe (TH), Germany
http://www.tm.uka.de/

Motivation

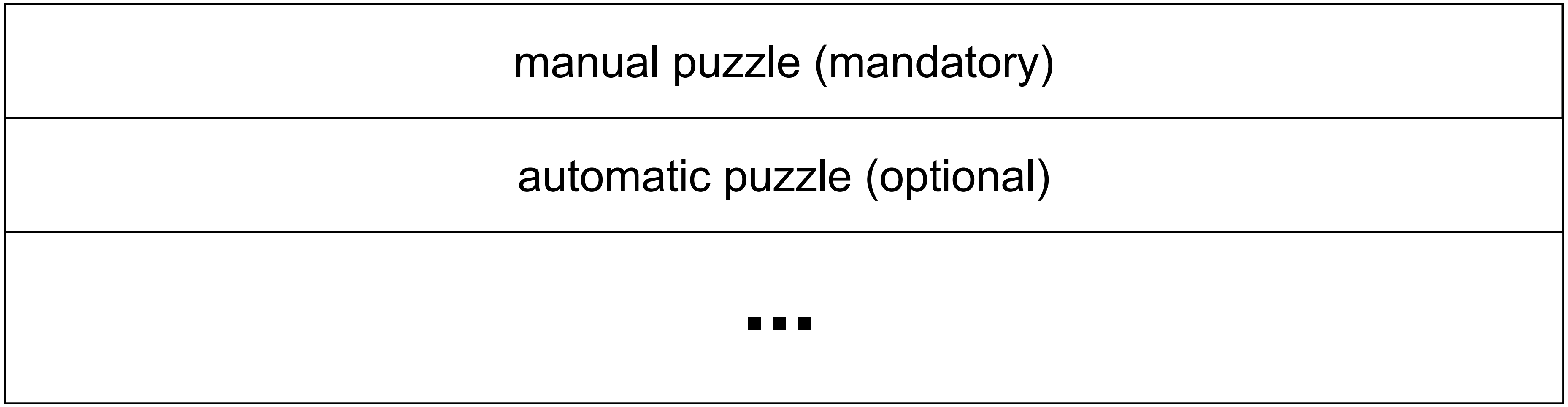
- Problem: Increasing number of spam mails
    - Sending spam is cheap (nearly no costs)
    - Spam often uses an existing but spoofed sender address
- Solution:
    - Effective verification of sender address
        - basically by using a Challenge/Response procedure
    - Increase costs of sending e-mails (by using a puzzle)
        - acceptable for a few e-mails
        - unacceptable for high-rate of automatically generated thousands of e-mails
        - Puzzle can be manually or automatically solved
        - Puzzle is not required for subsequent e-mails as sender is whitelisted when he solved the automatic puzzle
    - We do not claim that it is **the** Spam solution, but think it is viable and incrementally deployable, so it may be a starting point…

Basic Concept


sender
receiver
MUA
MDA
MTA
MDA
MUA
e-mail
e-mail
puts e-mail in **holdbox** (like a spambox) if sender is not whitelisted
puzzle mail
puzzle mail
puzzle mail
message gets „delivered“ (moved to INBOX) if puzzle solution correct
puzzle solution
puzzle solution
e-mail
sender of e-mail is put on a whitelist

- If sender is not on whitelist, e-mail gets only delivered to receiver (moved to INBOX) after sender solved a puzzle
  - A puzzle mail is created by receiver‘s MUA/MDA/MTA and sent to sender
  - Sender solves puzzle (if it belongs to a sent e-mail) and sends solution back to receiver‘s MDA

Puzzle Mail



manual puzzle (mandatory)

automatic puzzle (optional)

...

- Multipart MIME message containing at least two different puzzles:
  - manual puzzle solvable by a human, e.g., a picture with a number combination in it (Captcha - turing test).
  - automatic puzzle is a task which can be solved by a machine, e.g. to break a hash.
- Puzzles cause costs (user interaction or computation time) !!!
  - This would raise the cost for spammers dramatically
- Manual puzzles allow SAVE unaware users to send e-mails
- Automatic puzzles can be solved by instances of SAVE
  (e.g. MUA/MDA/MTA Plugin)

Optimizations/Extensions

- Integration into MDA/MTA
  - Enables automatic puzzle solving at e-mail provider
  - Independence of MUA (enables webmail)
  - Requires computation resources at provider
- Stateless Cookie Support
  - Prevents SAVE instances (MTA/MDA) to blindly solve puzzles
  - No additional state on sender’s MDA
  - Use of multiple MDAs/MTAs for load balancing possible
- Key Exchange prior to e-mail transfer
  - Subsequent mails are authenticated by H-MAC (e.g., hash over secret, nonce, To, From, Subject, Date, Msg-id and body)
    and hence do not require a puzzle check
  - Secret is bound to sender address, so sender cannot change address without reauthentication
  - Authenticated whitelist entries prevent whitelist guessing
    (and, whitelist entries may be aged)
  - Does only work if sender **and** receiver use SAVE

Special Cases

- e-mail forwarding
  - receiver's MDA uses forwarder address as puzzle sender address
  - transparent to forwarding
    (no additional functionality required at forwarding MTA!)
- mailing lists
  - manual addition of mailling list address to (unauthenticated) whitelist required (spammer must guess subscribed mailling lists of users to send spam), or,
  - provide secret when subscribing to mailing list (requires marginal extension to mailman, majordomo etc.)
    - mailing list s/w records secret in addition to normal subscriber data
    - mailing list s/w adds H-MAC to each mail that it forwards to the SAVE user
    - mailing list users will not be bothered by puzzle challenges
    - Spammers cannot use mailing list address as sender address

Advantages (1)

- **End-to-end** solution:
  - Incrementally deployable as MTA/MDA/MUA hook
  - No need to modify any in-transit MTAs, relays etc.
  - If used in MUA, providers are not burdened by additional computational effort
- **In-band** application layer solution works from end to end
  - no new or modified protocols in lower layers (e.g., SMTP) required
  - does not depend on other mechanisms or protocols like DNS/Web Servers
  - basically requires few X-headers or registered MIME types
- **Protects** receivers **immediately** on their own initiative
- User may peek into Holdbox for waiting mails from yet unknown/unauthenticated users

Advantages (2)

- Effectively verifies existence of sender address
  - If combined with Solicitation:-Header (RFC 4095/4096) could pursue spammers
- Classification solutions like SpamAssassin on receiving side will have decreasing work load with increasing SAVE deployment → May use saved CPU cycles for solving SAVE puzzles
- Puzzle complexity can be easily adapted to state-of-the-art
- Whitelist aging allows variable re-authentication intervals
- Allows to raise security incrementally as it gets more widely deployed
- Works effectively even if not deployed by other parties

(Potential) Disadvantages (1)

- Puzzles may annoy users if deployment increases
  - Install SAVE plugin and use Whitelist!
  - Different languages/users with disabilities may require variety of puzzles (Visual CAPTCHA, Audio CAPTCHA, ...)
- Spammers can still send Spam if they are willing to spend the effort, but…
  - Existence of sender address was verified, so spammer cannot masquerade, forge sender addresses
  - It is much more costly than now
- Does not prevent the **transmission** of spam (is delivered to HoldBox)
  - Mandatory puzzle interaction before e-mail transfer as next level of protection if widely deployed
- If used in MUA, user should stay online (few minutes) until potential puzzle comes back
  - not required for peers where sender is in whitelist already
  - If not possible (e.g., firewalled, send-only device), MDA plugin may help

(Potential) Disadvantages (2)

- Raised cost may not be relevant if spammers increase the number of zombies
  - But computational effort may be noticed by the legitimate owner of the zombie computer
  - SAVE-unware zombies are useless
  - Hi-jacked hosts can be detected by receivers if SAVE user gets “authorized” Spam
- Basic simple concept fails if spammers can guess whitelist entries
  - Should therefore use authenticated whitelist entries
- Reflection attack with amplification (also true for DNS, etc…)
  - Attacker sends small mail with spoofed addresses to SAVE user, which sends larger puzzle back (only b/w problem as puzzles will not be solved)

Comparison to Related Work

Penny Black Project

- Computational spam-fighting
- If a receiver implements Penny Black, all senders that are not whitelisted need to implement Penny Black, too.
- Ticket Server

HashCash

- no challenge/response

TMDA

- no puzzle
- no authenticated whitelist

SAVE

- User Interaction or computation for spam-fighting
- Fallback: If a senders MUA or MDA does not implement SAVE, the user can still solve the manual puzzle
- No server required