

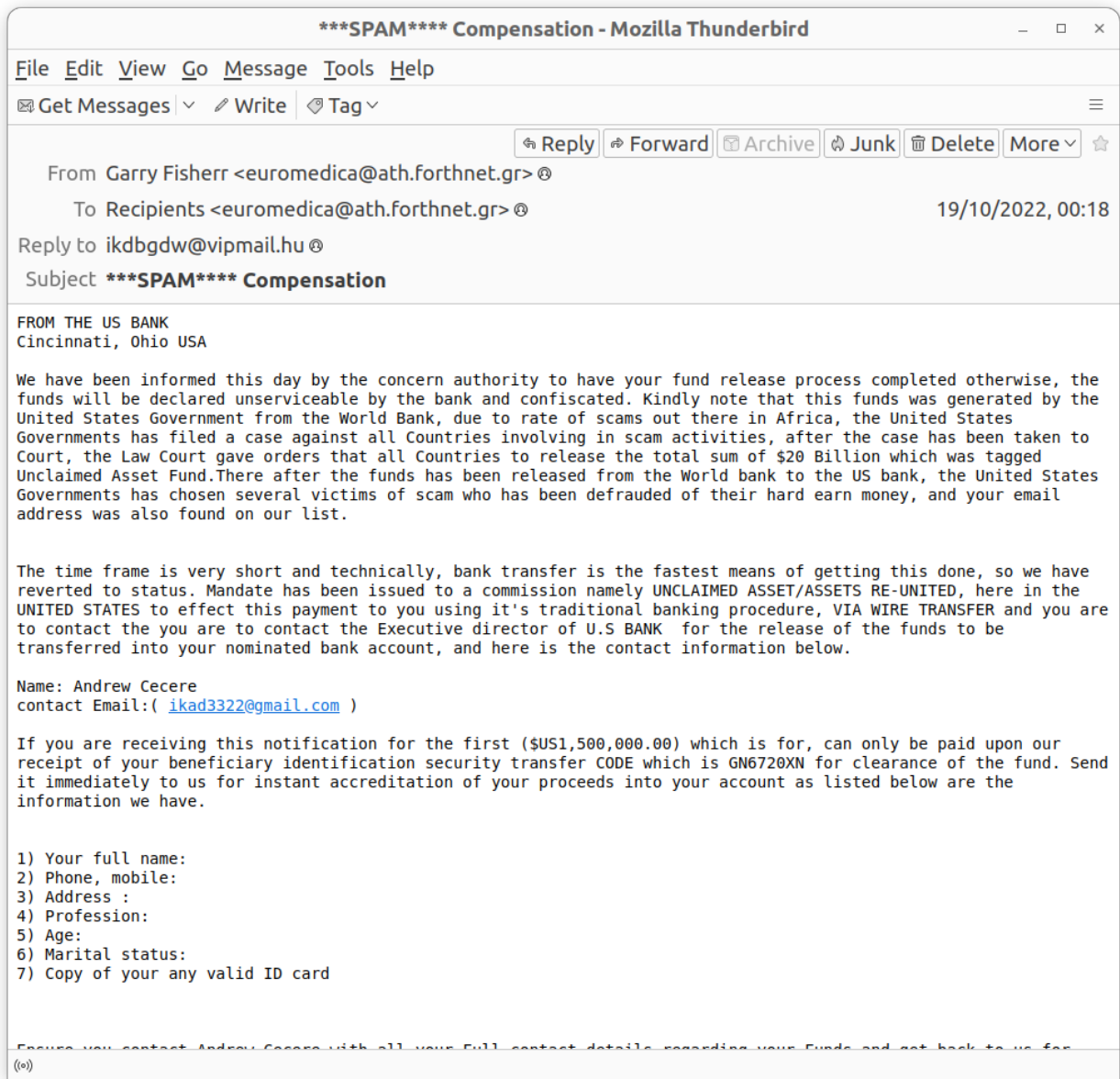
# TEU00311

What is the Internet doing to me?  
(witidtm)

Stephen Farrell  
[stephen.farrell@cs.tcd.ie](mailto:stephen.farrell@cs.tcd.ie)

<https://github.com/sftcd/witidtm>  
<https://down.dsg.cs.tcd.ie/witidtm>

# What do we think about this?



# Overview

- Email Architecture
- Spam
- End-to-end email security
- A few lessons to learn along the way

# Mail pre-history

- There used to be non-Internet email systems
  - uucp, X.400, other proprietary
- Internet email “won” - largely because it could connect all the others
- Fundamental basis of email is that anyone can (try) send mail to anyone, without needing permission first
  - Contrast with Twitter, FB etc. walled gardens
- Prime requirement: deliver the message, at all costs!
  - There really is no marginal cost:-)
  - That allows mail to be the basis for most other Internet/web services (e.g. so you can sign-up or do a password reset)
  - That also allows spam and other nasties
- Though there is no real per-message cost, running an effective mail server does have increasing costs
  - Often imposed by anti-spam measures and their ecosystem effects

# Mail Addresses

- Mail address: `user@example.com`
  - “user” is left-hand-side (LHS) of the “@”
  - Right Hand Side of the “@” is the email domain and is almost always a DNS domain
- RHS (plus DNS) is basically how mail gets routed to recipients – at example.com in the above
- LHS is (or should be) only interpreted at the recipient’s server(s)
- Internationalisation (I18N) for canonical mail addresses can be done but support isn’t so good, so we often see “decorated” forms of the email address:
  - “provost@tcd.ie” <notthat@example.com>
  - What you see in your mail user agent will vary
  - But it’s the bit in “<>” that counts – the decoration can be forged

# A helpful trick: LHS “+”

- For some mail services you can customise the LHS, e.g. if you have to give your email to some web site
- Instead of user@example.com you give them user+place@example.com
  - Other characters (not just “+”) can work sometimes
- Mail to user+place@example.com will still arrive in the mailbox for user@example.com
- That allows you to spam-folder them later and see who else got your address from them
- Can also avoid username collisions in services where email address is username and you want some form of separation, e.g. between work and personal accounts when forced to setup a web account somewhere
- Suggestion: try it out – doesn’t work for tcd.ie, but does for many mail providers though

# Use plaintext!

- If you want good email hygiene send plaintext email and render received emails as plaintext
- Why?
  - To locally avoid tracking pixels that are in almost all commercial email
  - Better interoperability
  - Much smaller attack surface for you (and to an extent, your correspondents)
- How?
  - Instructions at: <https://useplaintext.email>

# A recent battle...

- TCD started adding a “not from here” banner this year
- Mostly people do that by modifying the subject line
- TCD chose to add the banner to the message body, but did it wrong in a way that breaks things badly
- After much discussion, they stopped doing that for my account because I convinced them of the breakage
- I expect they’ll forget and I’ll have to fight that battle again
- They broke e2e email encryption (PGP & S/MIME) but the worst of it was how they handle “multipart/alternative” email bodies...



# The cause...

```
Content-Type: multipart/alternative; boundary="boundary-string"

--your-boundary
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

Plain text email goes here!

--boundary-string
Content-Type: text/html; charset="utf-8"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

<h1>This is the HTML Section!</h1>
<p>This is what displays in most modern email clients</p>

--boundary-string--
```

test.defo.ie/ifr x DEFO ECH Don x Michelle Gui x spasm x + - - x

mailarchive.ietf.org/arch/browse/spasm/

Mail Archive

Search www.ietf.org Search Datatracker Help Settings Sign In

Viewing List: spasm

<

Date

Thread

Search spasm

Q

Export

Subject	From	Date
[lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA	David A. Cooper	2024-08-01
[lamps] WG Last Call for draft-ietf-lamps-header-protection-23	Russ Housley	2024-08-01
[lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA	Markku-Juhani O. Saarinen	2024-07-31
[lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA	David A. Cooper	2024-07-31

9414 Messages

[lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA

"David A. Cooper" <david.cooper@nist.gov> | Thu, 01 August 2024 14:07 UTC | Show header

Yes, that is the correct OID for SHA-512. We do plan to specify in the FIPS the DER encodings of the OIDs of hash functions and XOFs that we think are most likely to be used. In addition to the sources that you mentioned, the OIDs for all of the NIST approved hash functions and XOFs are available at <https://csrc.nist.gov/projects/computer-security-objects-register/algorithm-registration>. This is also where the OIDs for ML-DSA, ML-KEM, and SLH-DSA will be posted.

The value for M' in the case of SHA-512 and an empty context string also seem correct.

On 7/31/24 1:48 PM, Markku-Juhani O. Saarinen wrote:  
> Thanks David. Somehow that got buried for me.  
>  
> The proposed formatting ( in <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/3KMh0D0pa30/m/vbflXo1xAQAJ>  
> ) seems sensible as the way it's specified there one doesn't need an  
> actual ASN.1 encoder for this, just hard-coded byte sequences for OIDs.  
>  
> If I was handed this spec, I'd guess that the OID would be "id-sha512"  
> defined in Appendix B of RFC 8017 (PKCS #1 v 2.2).  
> <https://www.rfc-editor.org/rfc/rfc8017.html#appendix-B.1>  
> <<https://www.rfc-editor.org/rfc/rfc8017.html#appendix-B.1>>  
>  
> id-sha512 OBJECT IDENTIFIER ::= {  
> joint-iso-itu-t (2) country (16) us (840) organization (1)  
> gov (101) csor (3) nistalgorith (4) hashalgs (2) 3  
>  
> }  
>  
> I'd absolutely specify the actual DER encoding in the FIPS spec ..  
> Unless I'm mistaken, in this case that would be 11 bytes (0x) 06 09 60  
> 86 48 01 65 03 04 02 03.  
>  
> This same OID (2.16.840.1.101.3.4.2.3) / id-sha512 already appeared 20  
> years ago in RFC 3560. In IETF specs this is wrapped in various ways

[lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA - Mozilla Thunderbird

File Edit View Go Message Tools Help

Get Messages Write Tag

Reply Reply List Forward Archive Junk Delete More

From David A. Cooper <david.cooper=40nist.gov@dmarc.ietf.org>

To Markku-Juhani O. Saarinen <mjos.crypto@gmail.com> 15:07

Cc Mike Ounsworth <Mike.Ounsworth@entrust.com>, pqc-forum <pqc-forum@list.nist.gov> MORE

Subject [lamps] Re: [pqc-forum] RE: Pre-hash options for ML-DSA

List-ID This is the mail list for the LAMPS Working Group <spasm.ietf.org>

[External Email] This email originated outside of Trinity College Dublin. Do not click links or open attachments unless you recognise the sender. Yes, that is the correct OID for SHA-512. We do plan to specify in the FIPS the DER encodings of the OIDs of hash functions and XOFs that we think are most likely to be used. I will post the DER encodings for ML-DSA, ML-KEM, and SLH-DSA will be posted.

The value for M' in the case of SHA-512 and an empty context string also seem correct.

On 7/31/24 1:48 PM, Markku-Juhani O. Saarinen wrote:  
Thanks David. Somehow that got buried for me.

The proposed formatting ( in <https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/3KMh0D0pa30/m/vbflXo1xAQAJ> ) seems sensible as the way it's specified there one doesn't need an actual ASN.1 encoder for this, just hard-coded byte sequences for OIDs.

If I was handed this spec, I'd guess that the OID would be "id-sha512" defined in Appendix B of RFC 8017 (PKCS #1 v 2.2). <https://www.rfc-editor.org/rfc/rfc8017.html#appendix-B.1>

id-sha512 OBJECT IDENTIFIER ::= {  
joint-iso-itu-t (2) country (16) us (840) organization (1)  
gov (101) csor (3) nistalgorith (4) hashalgs (2) 3  
}

I'd absolutely specify the actual DER encoding in the FIPS spec .. Unless I'm mistaken, in this case that would be 11 bytes (0x) 06 09 60 86 48 01 65 03 04 02 03.

This same OID (2.16.840.1.101.3.4.2.3) / id-sha512 already appeared 20 years ago in RFC 3560. In IETF specs this is wrapped in various ways which seem redundant. Avoiding more complicated formatting would be better.

One can try this out with OpenSSL asn1 parser with most Linux systems:  
\$ echo -en '\x06\x09\x60\x86\x48\x01\x65\x03\x04\x02\x03' | openssl asn1parse -inform der  
Which outputs  
0:d=0 hl=2 l= 9 prim: OBJECT :sha512

The proposed pre-hash sequence M' = octet(1) || octet(OLEN(ctx)) || ctx || OID\_PH || PH(M) would then typically (without ctx -- length 0) turn out to be in case of PH-SHA512

M' = (0x) 01 00 06 09 60 86 48 01 65 03 04 02 03 || SHA512(M)

right ?

Cheers,  
-markku  
Dr. Markku-Juhani O. Saarinen <[mjos@iki.fi](mailto:mjos@iki.fi)>

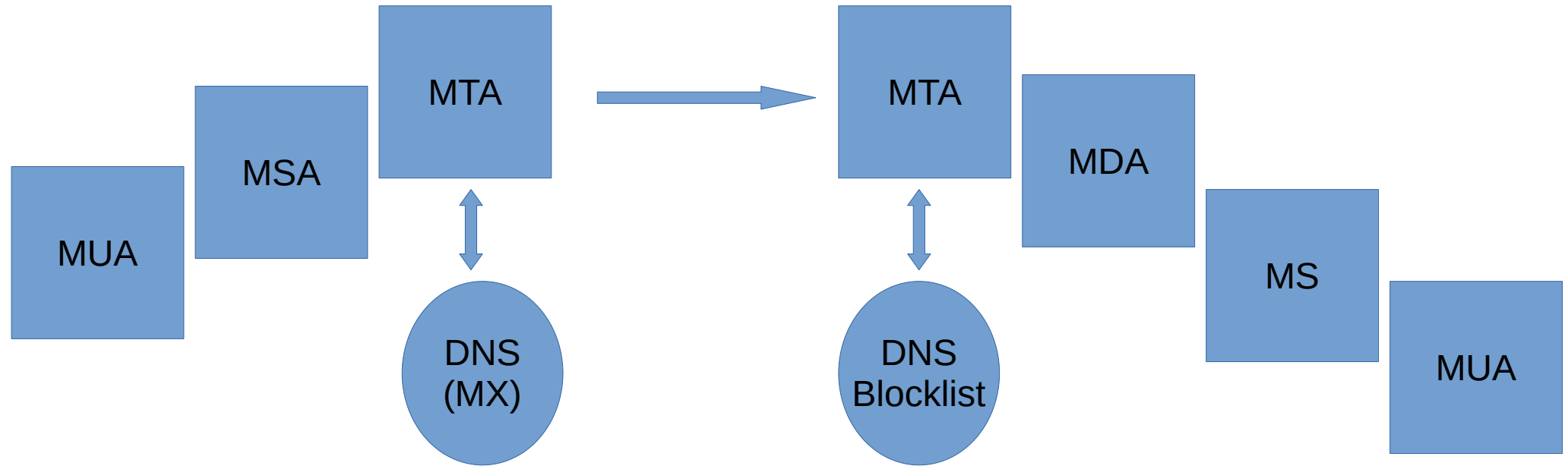
On Wed, Jul 31, 2024 at 7:44 PM David A. Cooper <[david.cooper@nist.gov](mailto:david.cooper@nist.gov)> wrote:  
The OIDs that we are discussing are the globally unique ASN.1 OIDs, such as would appear in the signature field of an X.509 certificate.

(0)

v2.26.8 | Report a Bug | By Email | System Status

# Email Architecture

- RFC5598 - <https://www.rfc-editor.org/rfc/rfc5598>



# Component Terminology

- As a prompt for me, so I cover the basics, we'll chat about various parts of the email ecosystem
- Mail User Agent (MUA) – that's the “client” you use, which could be outlook, thunderbird, a phone-app, or a Web User Agent, where you access your mail via a normal web browser
- Mail Submission Agent (MSA) – the thing to which email is submitted by an MUA, often via the SMTP or SMTP/SUBMIT protocol
- Mail Transmission Agent (MTA) – a server that receives mail routes it to the next MTA on it's way to the recipient, and that can do other things, e.g. scan for spam/malware using the SMTP protocol, hopefully, and often, over TLS so the mail is encrypted **in-transit**
- Mail Delivery Agent (MDA) – mail delivery agents delivery of email into a message store or generating a bounce message e.g. if no such mailbox exists
- Message Store (MS) – is where your email messages live before (and after) you read 'em and is how you can see the same messages from different MUAs (e.g. laptop/phone)

# Protocol Terminology

- Protocols for fetching messages from MS to MUA:
  - Internet Message Access Protocol (IMAP)
  - Earlier similar thing: Post office Protocol (POP/POP3)
- Protocol for sending messages from MUA to MTA, or between MTAs: Simple Mail Transfer Protocol (SMTP)
- Today: all of those should really be run over TLS, which is true for about 99% of messages
  - If not, you're using the wrong service!
- Not all email interactions are standardised (actually a significant %):
  - Web MUA<->MS protocol generally proprietary stuff done via Javascript
  - Outlook/MS-Exchange also does proprietary stuff

# Mailing Lists

- Mailing lists are (mostly) handled via special MUAs – you send to the list address and then a list server creates new email messages (with the same content) and sends those to a set of list subscribers
- Might seem v. basic, but that is how most Internet standardisation still happens!

# Message Terminology

- Message = envelope + body
- Envelope = sequence of header fields:
  - E.g. From:, Date:, Subject:, Resent-From: ...
- Body = set of MIME body-parts
  - MIME = Multimedia Internet Media Extensions is how cat images are added to emails, which were originally just ASCII bodies

# Message Attachments

- Messages can have attachments, e.g. images, ms-word files, pdfs
- Those can trigger applications, e.g. if you click to open the attachment
- Some of those can be dangerous, e.g.
  - ms-word files can contain macros (small programs) that might be run when you open the file – that should be turned off (I'm not sure what the default is nowadays with “cloudy” office licensing etc.)
  - An attachment might be a specially crafted file that triggers a bug in a common application used to open that file format – that's a typical way to attempt to distribute malware
  - An attachment might contain an “executable” file – a program that you can run – you'll be warned but it's easy to go wrong on when having a bad day
- When sending: try send the least dangerous format attachment, often that'd be a PDF or raw image file (JPEG, PNG)
- When receiving: try setup your MUA to be more conservative and don't just open random attachments even if they're from people you know



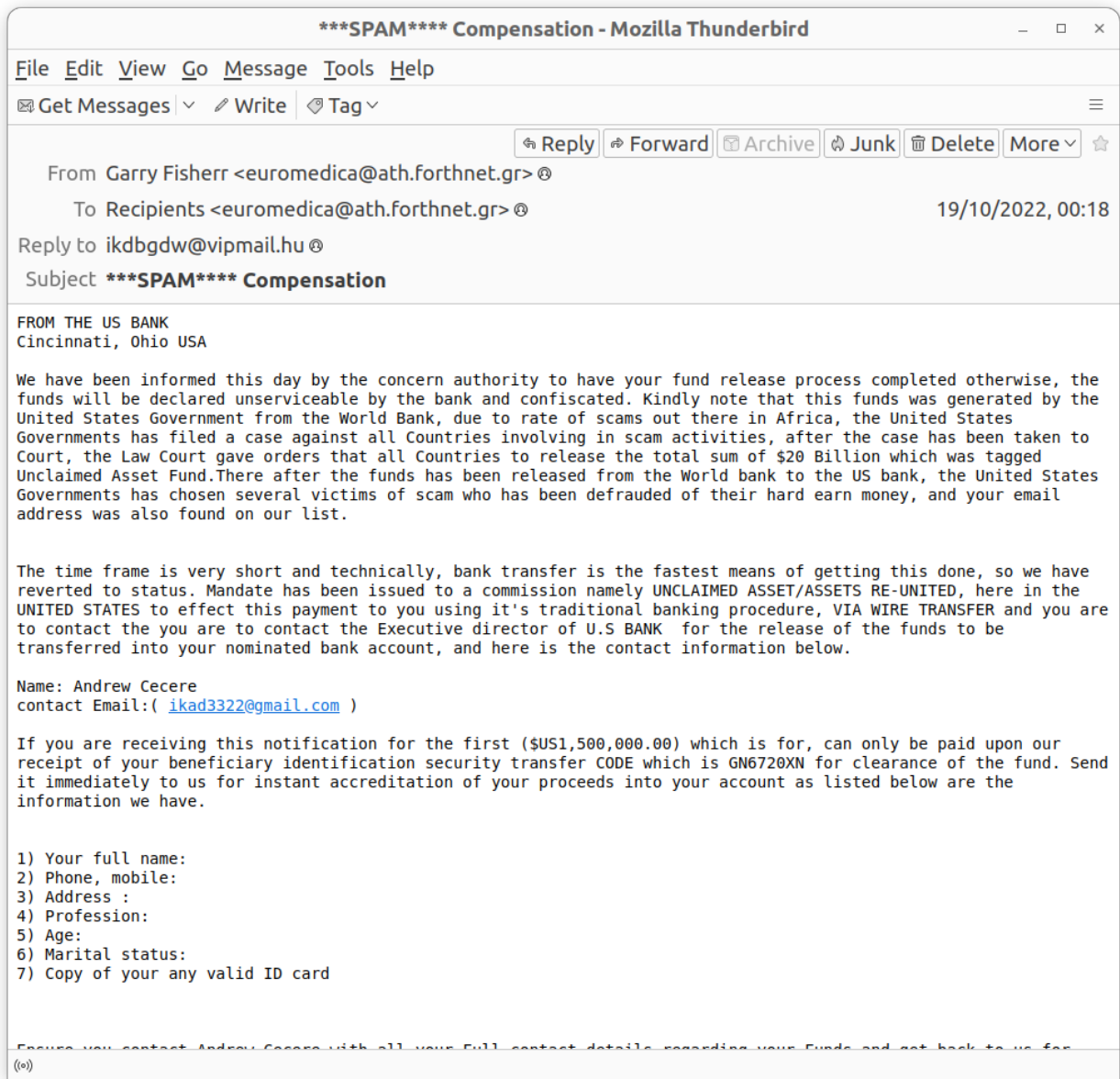
# Spam



A container of Spam rests at 4,947 meters on the slopes of a canyon leading to the Sirena Deep in the Mariana trench. Photograph: Noaa Office of Ocean Exploration

<https://www.theguardian.com/environment/2017/feb/13/extraordinary-levels-of-toxic-pollution-found-in-10km-deep-mariana-trench#img-1>

# What do we think about this?



# Advance fee fraud

- “We have been informed this day by the concern authority to have your fund release process completed otherwise...”
- Method:
  - 100: I’ll send you €1M once you send me €1
  - 200: Receive €1
  - 300: GOTO 100
- Why is the language in such emails often poor?
- Don’t be disparaging of those who fall for fraud! We’ll all have a bad day some day.

# What is spam?

- Various acronyms:
  - Unsolicited bulk email (UBE)
  - Unsolicited commercial email (UCE)
- Spam is bad:
  - Would overwhelm utility of mail if not countered
  - Resource consumption
    - Filters, scanners etc. cost time & money
  - Malware
  - Phishing attempts

# Original spam tricks

- Just send email!
  - Ahh...the naivety of it all!
- Email to list:
  - List servers got better, e.g. subscriber only with controlled subscription
- Forge headers and submit to your own server
  - Mostly outbound messages are sanity checked now
- Send via open relay
  - Used to be a lot of these, very few now

# Originating Spam

- Rent a botnet for 48 hours, send some millions of spam
  - Ideally using the mail credentials of the botted device's true owner
  - Georgoulas, Dimitrios, et al. "Botnet business models, takedown attempts, and the darkweb market: A survey." ACM Computing Surveys 55.11 (2023): 1-39. APA  
[https://backend.orbit.dtu.dk/ws/portalfiles/portal/289728739/Survey\\_on\\_Botnet\\_Economics\\_3\\_.pdf](https://backend.orbit.dtu.dk/ws/portalfiles/portal/289728739/Survey_on_Botnet_Economics_3_.pdf)
- Can be done via single compromised hosts but those'll quickly be closed down
- Spend a few quid, setup a supposedly "legitimate" domain, wait a month or two, then send millions of spams before you're shut down
  - Cost: maybe O(10k€) @ high-end, but might still make a profit
- State of the art in originating spam will always be ahead of defenders so long as there's money to be made

# Sometimes hard to know...

- Is this spam?

“HILARY TERM GREETINGS FROM THE COLLEGE CHAPLAINS The College Chaplains send best wishes to all, and would like to bring the following upcoming events to your attention. They are open to any students or staff members who wish to join us. ...”

- There are “legitimate” bulk email senders – for announcements, marketing, support, newsletters,...
- Those can easily start to look like spam (esp. marketing)

# More spam tricks...

- Confusion:
  - accounts@paypa1.com
  - support@eboy.com
  - postmaster@boi-support.com
  - About to get worse thanks to l18N
  - security@bigbank.com
    - ^ Unicode 0430 is cyrillic small 'a'
- Throwaway domains/addresses
- Zombie hosts
- Trojans
- Fake ISPs



# HTML messing

- Originally mail was text based, then we added MIME, one kind of MIME body is “text/html” which turns your MUA into a dangerous web browser
- Vulnerabilities created by HTML rendering in your MUA:
  - Colour-related trickery
  - Font size 0: break words with zero width spaces
  - Presentation of nice-looking link-text not clearly-dodgy URL
- Web-bugs: use of image URL that includes a value related to you allows sender to track when you open a mail if you render images

# Solution to HTML messing

- Solution: turn off images and all HTML rendering in your MUA!
- While there: turn off clever fonts too if you can
- Both can be tricky on phones – in that case turn it off on your laptop and wait to read possibly dodgy mails there
- You do NOT have to buy into other people's idea of what is urgent!

# How much spam is there?

- Lots
  - Hard to get good figures, these are ones I've overheard
- ISP backbones:
  - 70% + of email traffic
- Delivered mail:
  - 40% + delivered
- Increasing or not?
  - Harder to tell if MTAs silently filter

# Some Anti-spam techniques

- Content filtering (Bayesian, etc.)
- DNS Block Lists (SORBS, DNSBL)
- Register of known spam operators (ROKSO)
- Greylisting
- Sender Policy Framework (SPF)
- Domain Keys Identified Mail (DKIM)
- Domain Message Authentication Reporting and Conformance (DMARC)

# End-to-end (e2e) email security

- That's where you use cryptography to protect messages between the sender and recipient
  - Rather than just between each “hop” on the path the mail follows
- There are two different standards for how to do that: S/MIME and PGP
  - S/MIME is mostly used inside enterprises/govt
  - PGP is mostly used by nerds or smallish groups

# Deployment

- Most MUAs support s/mime or PGP either built-in or as an option
  - There are also “plug-in” products
- And mostly then *can* work together
  - I’ve used both, PGP more usable (via Thunderbird/Enigmail)
- But secure mail is not ubiquitous
  - Why?

# e2e email security barriers

- Designs pre-date web user agent which changes trust model (where's the private key kept? Needs new infrastructure)
- Needs all major email service providers (yahoo, hotmail, gmail) to deploy the same thing which also needs to be implemented by all major user agent developers (microsoft, mozilla, apple, google)
- Public key retrieval needs to be fixed (doable if the above done, but a killer if not done), likely with some new PKI (doable but who's gonna pay?)
- Mail headers need to be protected as users don't get that S/MIME and PGP only protect body and not e.g. Subject, From (new enveloping protocol needed, can be done but kludgy)
- We need to unify S/MIME and PGP or pick one or we'll lose interop (it's ok if the other soldiers on for some niches)
- Users don't care much, so it has to be entirely transparent for them (needs significant UI work, co-ordinated across MUAs and significant web-UAs)

# What can you do?

- Target diversity – don't all use the same service(s)
- Use LHS “+” trick if it works for your service
- Don't react immediately
- Take **much** longer to write mails than to read mails
  - Always re-read before you send
- Don't render mails as HTML if your MUA allows that
  - It's ok to wait until you're using a laptop to process mail
- Don't assume any names/links displayed to you are real
- Treat all mails and especially attachments with caution