Introduzione ai remailer anonimi

Posta elettronica anonima

Versione 2.1

frx@firenze.linux.it

PWS — Progetto Winston Smith

FLUG — Firenze Linux User Group

Sommario

- Cos’è un remailer anonimo?
- A cosa serve? Perché è importante?
- Come funziona?
  - Tipo 0
  - Tipo I
    ↪ Reply-block e nym server
  - Tipo II
  - Tipo III
- Conclusioni

Cos’è un remailer anonimo?

C’è anonimato in rete?

Si sente dire spesso:

> *“Su **Internet** si riesce facilmente a raggiungere e mantenere l’anonimato”*

Sbagliato!

In realtà, **senza opportuni accorgimenti**, su Internet siamo tutti rintracciabili e sorvegliabili (molto di più che con i mezzi di comunicazione ‘tradizionali’!).

La posta elettronica?

Ogni messaggio e-mail accumula durante il suo tragitto la traccia dei nodi attraversati:

```
Return-Path: <laura@azienda.it>
Received: from smtp.isp.it (194.71.2.5) by ims4a.isp.it (6.0.013)
 id 3EDE7DF00945 for piero@isp.it; Sun, 8 Jun 2003 15:21:51 +0200
Received: from mail.org.it (194.111.3.1) by smtp.isp.it (6.0.012)
 id 3EDC132DDE17 for piero@isp.it; Sun, 8 Jun 2003 15:21:51 +0200
Received: by mail.org.it (Postfix)
 id 108933C531; Sun,  8 Jun 2003 15:19:12 +0200 (CEST)
Delivered-To: piero@org.it
Received: from smtp.azienda.it (unknown [211.235.223.197])
 by mail.org.it (Postfix) with ESMTP id 018603C50E
 for <piero@org.it>; Sun,  8 Jun 2003 15:18:12 +0200 (CEST)
Received: from vega (vega.station.lan [10.2.0.20])
 by smtp.azienda.it (Postfix) with ESMTP id D56312DC0F
 for <piero@org.it>; Sun,  8 Jun 2003 15:20:47 +0200 (CEST)
```

Remailer anonimi

Un **remailer anonimo** (*anonymous remailer*) è

> un sistema che permette ai suoi utenti di inviare messaggi e-mail anonimi

Nel messaggio che arriva al destinatario non ci sono tracce dell'identità del vero mittente: le **intestazioni** 'sensibili' sono state **rimosse** o sostituite. Inoltre il remailer non mantiene alcun tipo di registro (**log**).

A cosa serve? Perché è importante?

Anonimato forte

Remailer $\Rightarrow$ effettiva irrintracciabilità anche da parte di organizzazioni potenti e determinate

> *"L'anonimato serve solo ai terroristi e ai pedofili."*
>
> *"Perché una persona onesta dovrebbe desiderare l'anonimato?"*

Per difendere la sua libertà di parola!

Chi vuole essere anonimo?

- persone che divulgano informazioni scomode per i politici o per le multinazionali
- vittime di violenze domestiche
- vittime del racket di un'organizzazione mafiosa
- dissidenti politici in regimi oppressivi
- vittime di intolleranza religiosa, sociale o politica

Chiaramente sono possibili anche usi 'cattivi', ma il remailer è solo uno strumento.

Ancora perplessi?

Chiunque è pronto a difendere il diritto all'**identità**.

Il diritto all'**anonimato** è il suo *duale* (diritto a non rivelare la propria identità).

> *"I remailer servono agli spammer?"*

No!

- mezzo intrinsecamente lento
- contromisure (uso di blacklist, cancellazione di grandi quantità di messaggi simili in senso bayesiano, . . .)

Come funziona?

Funzionamento dei remailer

- **Tipo 0** (remailer pseudoanonimi, anche detti remailer di tipo “penet”): bassa sicurezza, ormai dismessi
- **Tipo I** (remailer cypherpunk): sicurezza elevata se usati in catena e con crittografia
- **Tipo II** (remailer mixmaster): sicurezza ancora più elevata, possono funzionare anche in modalità cypherpunk
- **Tipo III** (remailer mixminion): tipo sperimentale attualmente in sviluppo

Tipo 0

Remailer pseudoanonimi

Remailer di **tipo 0** (detti anche remailer “penet” da `anon.penet.fi`):

- forniscono all’utente uno pseudonimo casuale (come `11891@rem0.com`)
- mantengono una base dati segreta di corrispondenze pseudonimo↔utente
- i messaggi (pseudo)anonimi sembrano provenire da un account di posta sul remailer
- i messaggi indirizzati ad uno pseudonimo vengono passati all’utente reale

Debolezza dei remailer di tipo 0

Esiste un archivio delle identità reali degli utenti:

- attacco mediante intrusione nella macchina ospite
- attacco mediante estorsione dei dati riservati all'amministratore

Successe nel 1995 al remailer `anon.penet.fi` (attivo dal 1993), il cui amministratore (Johan "Julf" Helsingius) fu costretto dalla magistratura a rivelare la vera identità dell'utente `an144108`

Storia di anon.penet.fi

- nel 1995 la *Chiesa di Scientology* denuncia l'utente `an144108@anon.penet.fi` che ha divulgato testi interni di questa setta californiana
- Julf Helsingius viene sentito come testimone
- la polizia gli chiede di rivelare l'intero archivio
- lui rifiuta e così nasce una causa legale
- alla fine Julf non consegna l'intera base dati, ma viene costretto a rivelare l'identità di `an144108`
- `anon.penet.fi` rimane attivo ancora un po' e poi chiude per decisione di Julf

Vedi `http://www.december.com/cmc/mag/1997/sep/helmers.html`

Tipo I

Remailer cypherpunk

Remailer di **tipo I**:

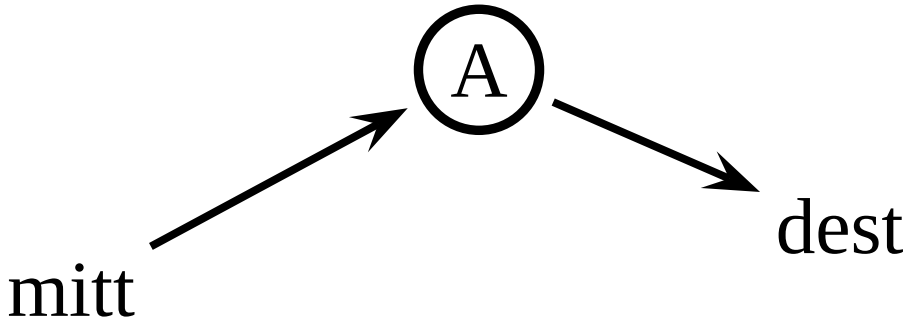
- sono più sicuri dei remailer pseudoanonimi
- non hanno basi dati di utenti, né registri (log)
- i messaggi anonimi sembrano tutti provenire dallo stesso utente (fittizio) del remailer
- i messaggi inviati all'utente fittizio rimbalzano
- non richiedono programmi particolari, lato utente (sono sufficienti un editor di testo, un MUA e un programma OpenPGP)
- esistono in alternativa dei client che automatizzano la procedura…

Usare un remailer cypherpunk

```
==========================================
From: mitt@origine.it
To: remailer@a.org
Subject: Buone notizie
Content-Type: text/plain; charset=us-ascii
------------------------------------------
::
Request-Remailing-To: dest@porto.it

Messaggio anonimo.
==========================================
```

oppure
Anon-To:



Cosa arriva a dest?

```
==========================================
Return-Path: nessuno@a.org
From: nessuno@a.org
To: dest@porto.it
Subject: Buone notizie
------------------------------------------
Messaggio anonimo.
==========================================
```

- le intestazioni ‘sensibili’ sono state rimosse
- il messaggio sembra provenire da un indirizzo fittizio
- il subject è stato conservato (non garantito)

Debolezza: messaggio in chiaro

Problema: il messaggio che `mitt` invia al remailer è in chiaro

$\Rightarrow$ **Attacco:** chi intercetta la posta spedita da `mitt` può scoprire contenuto, destinatario e vero mittente del messaggio anonimo

Contromisura: `mitt` può usare la crittografia a chiave pubblica (OpenPGP) per inviare il messaggio al remailer

N.B.: per questo motivo, molti remailer non accettano messaggi in chiaro

Cypherpunk con OpenPGP (1)

1. `mitt` prepara un file di testo come questo:

```
--------------------------------------------
::
Request-Remailing-To: dest@porto.it

Messaggio anonimo.
--------------------------------------------
```

2. lo cifra (GnuPG) con la chiave pubblica **del remailer**
3. salva l'output (ASCII-armored) su un secondo file

Cypherpunk con OpenPGP (2)

4. vi antepone la direttiva `Encrypted: PGP`

```
-----------------------------------------
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.6 (GNU/Linux)

hQEOAyH+c/4I3117EAP/T7JoDNFqfbmLn396kb4OdaQB
j8TTpjjOttoRAeKxejkbPOBRWrC+dKUtYQ6B6zxfmgNJ
ISBZZvW24Y00ssbTLAHkkKE=
=z/t2
-----END PGP MESSAGE-----
-----------------------------------------
```

Cypherpunk con OpenPGP (3)

5. spedisce il tutto al remailer

```
==========================================
From: mitt@origine.it
To: remailer@a.org
Subject: Buone notizie
------------------------------------------
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----

[...]

-----END PGP MESSAGE-----
==========================================
```

Debolezza: subject in chiaro

Problema: il subject del messaggio che `mitt` invia al remailer è in chiaro

$\Rightarrow$ **Attacco:** chi intercetta i messaggi che arrivano al e partono dal remailer può correlarli in base al subject

Contromisura: `mitt` può inserire il subject nella parte cifrata

N.B.: alcuni remailer rimuovono il subject $\Rightarrow$ specificarlo come direttiva può essere l'unico modo per ottenerne uno

Cypherpunk: comando

Serve per specificare intestazioni aggiuntive che il remailer inserirà nel messaggio anonimo.

```
--------------------------------------
::
Request-Remailing-To: dest@porto.it

##
Subject: Buone notizie

Messaggio anonimo.
--------------------------------------
```

Il subject del messaggio da inviare al remailer può essere lasciato vuoto.

Debolezza: invio immediato

Problema: il messaggio anonimo viene spedito dal remailer poco dopo l'arrivo del messaggio di `mitt`

$\Rightarrow$ **Attacco:** chi può monitorare il traffico di posta (in entrata ed in uscita) del remailer può correlare i messaggi in base ai tempi di arrivo e partenza

Contromisura: *reordering*

Cypherpunk: reordering

- funzionalità implementata in tutti i remailer cypherpunk (moderni)
- operazione effettuata automaticamente
- l'invio viene ritardato di un intervallo casuale $\Delta t \leqslant \Delta t_{\text{max}}$ (detto "**latency**")

  $\Rightarrow$ sufficiente in caso di alto traffico
- n messaggi vengono trattenuti sul remailer; quando arriva un ulteriore messaggio, viene inviato un messaggio anonimo scelto casualmente tra quelli del "**pool**"

  $\Rightarrow$ ulteriore protezione in caso di basso traffico

Debolezza: solo reordering?

Problema: il pool è comunque limitato

$\Rightarrow$ **Attacco:** chi può inviare al remailer un numero enorme di messaggi (*spam attack*) rinnova completamente il pool e riconosce il messaggio anonimo in mezzo ai suoi, a lui noti (*blending attack*)

Contromisura: direttiva `Latent-Time:`

Cypherpunk: Latent-Time

Serve per specificare un ritardo aggiuntivo (indipendente dal reordering) durante il quale il messaggio anonimo viene comunque trattenuto.

```
-----------------------------------------
::
Request-Remailing-To: dest@porto.it
Latent-Time: +6:00

##
Subject: Buone notizie

Messaggio anonimo.
-----------------------------------------
```

In questo caso il messaggio viene trattenuto per 6 h

Sintassi di Latent-Time

`Latent-Time: +4:20`

$\Rightarrow$ il messaggio viene trattenuto per 4 h + 20 min (non si può richiedere più di 24 h)

`Latent-Time: +5:15r`

$\Rightarrow$ il messaggio viene trattenuto per un intervallo di tempo casuale non superiore a 5 h + 15 min

`Latent-Time: 19:50`

$\Rightarrow$ il messaggio viene trattenuto fino alle ore 19:50, ora locale del remailer

Debolezza: lunghezza nota

Problema: dalla lunghezza del messaggio che `mitt` invia al remailer si può prevedere la lunghezza del messaggio anonimo corrispondente e così riconoscerlo

$\Rightarrow$ **Attacco:** chi può monitorare il traffico di posta (in entrata ed in uscita) del remailer può correlare i messaggi in base alle loro lunghezze

Contromisura: direttiva `Cutmarks:`

N.B.: contromisura parziale

Cypherpunk: Cutmarks

Serve per specificare un marcatore di fine messaggio anonimo: ciò che segue viene ignorato.

```
-----------------------------------------
::
Request-Remailing-To: dest@porto.it
Latent-Time: +6:00
Cutmarks: --

##
Subject: Buone notizie

Messaggio anonimo.
--
Spazzatura che non comparira‘ nel messaggio...
-----------------------------------------
```

Difetti del Cutmarks

Grazie alla tecnica `Cutmarks:`, il messaggio anonimo sarà più corto del previsto.
Tuttavia:

- il messaggio che `mitt` invia al remailer potrebbe crescere al di sopra dei limiti di accettazione
- è una tecnica poco usata $\Rightarrow$ il messaggio potrebbe essere l'unico a presentare una diminuzione di lunghezza diversa da quella prevista

N.B.: sarebbe meglio se tutti i messaggi avessero la stessa lunghezza

Debolezza: remailer onesto?

Problema: il remailer conosce contenuto (se non cifrato con la chiave pubblica del destinatario), destinatario e vero mittente del messaggio anonimo; ci fidiamo?

$\Rightarrow$ **Attacco:** chi ha compromesso il remailer (all'insaputa dell'amministratore) oppure l'amministratore (se è in mala fede e ha configurato il remailer in modo da 'loggare' tutto il traffico) può scoprire tutto

Contromisura: usare una catena di remailer

Cypherpunk in catena (1)

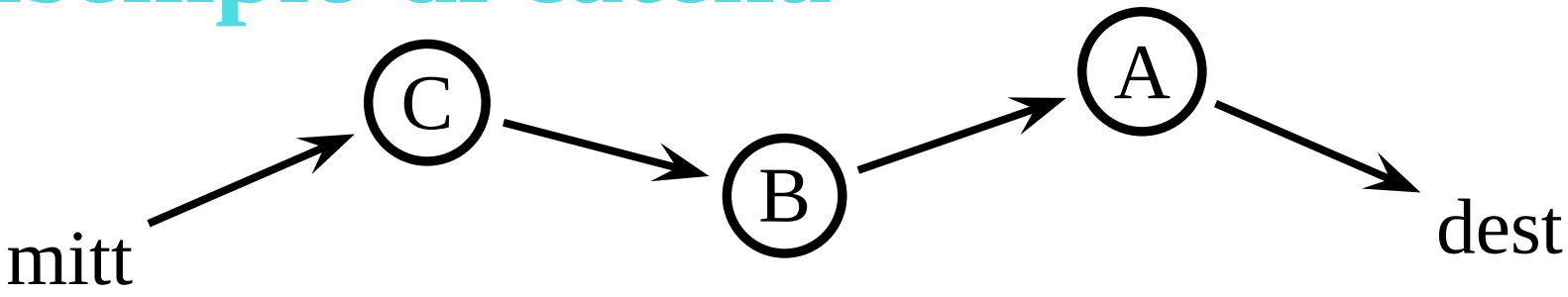
Abbiamo visto come `mitt` può inviare un messaggio anonimo a `dest` attraverso il remailer A:

1. `mitt` prepara un file di testo con il corpo del messaggio anonimo
2. vi antepone la direttiva di remailing a `dest` ed altre opportune direttive per il remailer A
3. cifra il file con la chiave pubblica del remailer A
4. antepone la direttiva `Encrypted: PGP`
5. il risultato è il corpo di un messaggio da spedire al remailer A

Cypherpunk in catena (2)

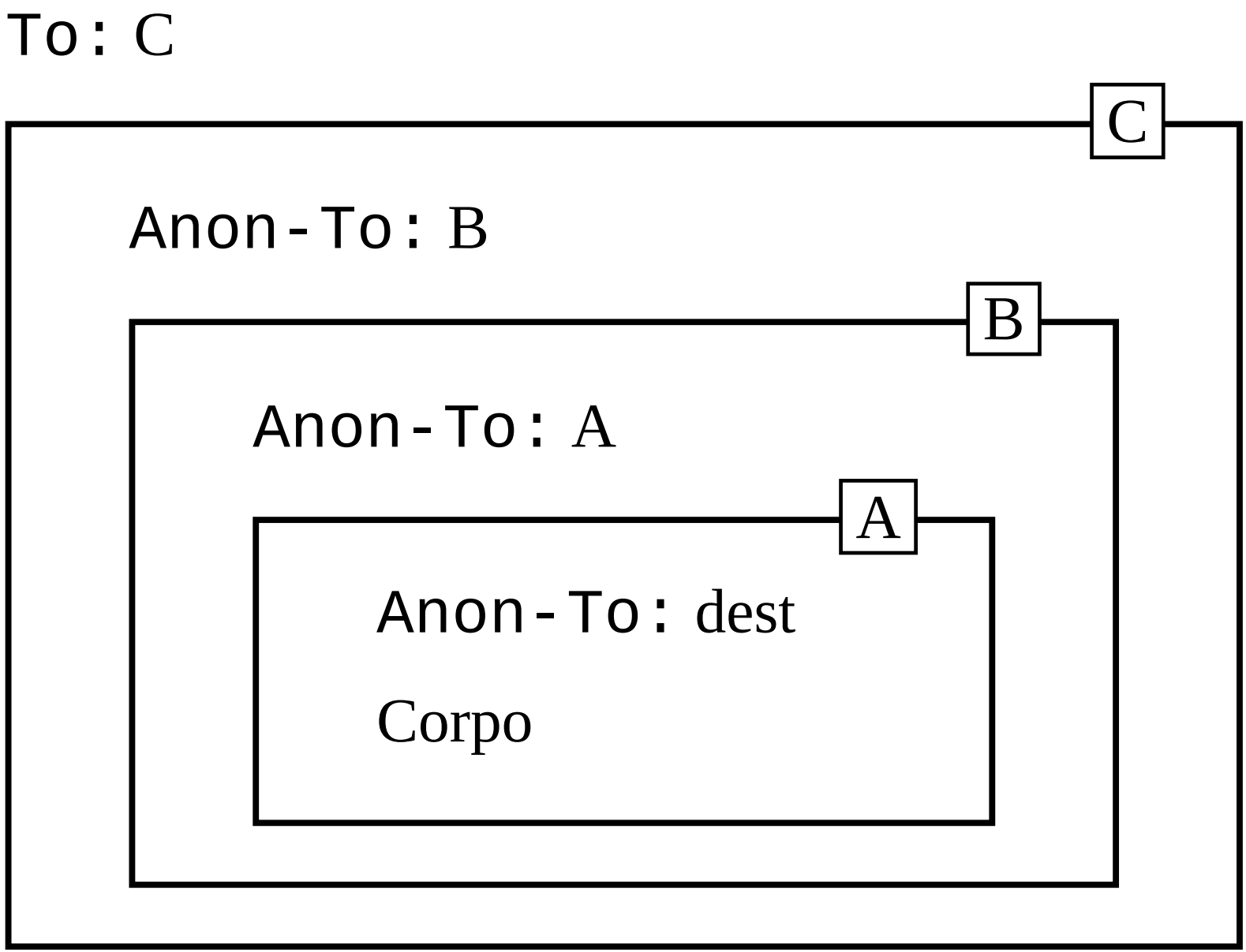
- Quindi, alla fine della preparazione, `mitt` deve spedire un messaggio al remailer A
- può farlo attraverso un altro remailer (B)!
- partendo dal corpo del messaggio da spedire al remailer A, `mitt` ripete la procedura (con B al posto di A ed A al posto di `dest`)
- ottiene così un messaggio da inviare al remailer B
- può farlo attraverso un terzo remailer (C)!
- …

Esempio di catena



- C sa che `mitt` richiede l'invio di un messaggio anonimo a B, ma non conosce il contenuto del messaggio (solo B può decifrarlo)
- B sa che qualcuno (attraverso C) richiede l'invio di un messaggio anonimo ad A, ma non conosce il contenuto del messaggio, né il vero mittente
- A sa che qualcuno (attraverso B) richiede l'invio di un messaggio anonimo a `dest`, ne conosce il contenuto, ma non il vero mittente

Messaggio cifrato a “cipolla”



Efficacia della concatenazione

- per ricostruire il percorso `mitt` $\rightarrow$ `dest`, è necessario che **tutti** i remailer della catena siano compromessi

  $\Rightarrow$ l'anonimato è ottenuto, purché **almeno un** remailer della catena sia sicuro

- incrementando la lunghezza della catena, si può aumentare la probabilità di trovare un remailer sicuro
- si consiglia una catena di almeno **quattro** remailer (non molti di più per motivi pratici)

Debolezza: traffico analizzabile

Problema: i remailer non possono nascondere il fatto che ricevono e spediscono posta

$\Rightarrow$ **Attacco:** chi può

1. monitorare il traffico di posta in ampie porzioni della rete e
2. inviare al remailer un numero enorme di messaggi identici a quello da tracciare

(*replay attack*), può ricostruire il percorso rilevando un improvviso e anomalo aumento del traffico tra gli elementi della catena

Contromisura: nessuna (con i remailer cypherpunk)

N.B.: sarebbe meglio se i remailer si rifiutassero di inviare più di una volta lo stesso messaggio

Reply-block e nym server

Cypherpunk: risposte?

- con l'introduzione dei remailer cypherpunk, `dest` perde la possibilità rispondere a `mitt`
- la recupera se `mitt` fornisce un "**reply-block**" (una specie di ricetta cifrata per raggiungerlo attraverso una catena di remailer)

  $\Rightarrow$ `dest` deve eseguire manualmente e con precisione alcune operazioni scomode
- i **nym server** automatizzano il processo nascondendo il reply-block dietro un *nym*

Cypherpunk: reply-block (1)

`mitt` prepara un messaggio cypherpunk **privo di corpo** indirizzato **a se stesso** attraverso una catena di remailer a sua scelta:

1. `mitt` crea un file di testo come questo:

```
-----------------------------------------
::
Request-Remailing-To: mitt@origine.it
Latent-Time: +4:00r
-----------------------------------------
```

2. lo cifra con la chiave pubblica dell'ultimo remailer della catena
3. antepone la direttiva `Encrypted: PGP`

Cypherpunk: reply-block (2)

4. antepone le direttive per il penultimo remailer

```
-----------------------------------------
::
Request-Remailing-To: remailer@ultimo.org
Latent-Time: +3:11

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----

[...]

-----END PGP MESSAGE-----
-----------------------------------------
```

Cypherpunk: reply-block (3)

5. cifra con la chiave pubblica del penultimo remailer
6. antepone la direttiva `Encrypted: PGP`
7. e così via fino al primo remailer…

In questo modo ottiene il reply-block:

```
--------------------------------------------
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
[...]
-----END PGP MESSAGE-----
--------------------------------------------
```

Cypherpunk: reply-block (4)

Nel messaggio anonimo che `mitt` invia a `dest` saranno presenti le istruzioni per rispondere:

[...]

```
Per rispondermi, copia il blocco di testo che
trovi qua sotto e inseriscilo all'inizio della
risposta.
Invia il messaggio a <remailer@primo.org>

---+--+--- da qui ---+--+--- da qui ---+--+---
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
```

[...]

```
-----END PGP MESSAGE-----
```

Debolezza: corpo in chiaro

Problema: il corpo della risposta di `dest` è in chiaro

⇒ **Attacco:** chi intercetta i messaggi che arrivano al e partono dai remailer può correlarli in base alla parte non cifrata

Contromisura: `mitt` può richiedere una cifratura simmetrica diversa ad ogni passaggio:

```
----------------------------------------
::
Request-Remailing-To: mitt@origine.it
Latent-Time: +4:00r
Encrypt-Key: ultima_password
----------------------------------------
```

N.B.: non tutti i remailer cypherpunk supportano questa funzionalità

Cypherpunk: Encrypt-Key

Tutto ciò che sta al di sotto del marcatore ** viene cifrato con la chiave simmetrica specificata ⇒ Occorre aggiungere il marcatore alla fine del reply-block (dopo una riga vuota):

```
----------------------------------------
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
[...]
-----END PGP MESSAGE-----

**
----------------------------------------
```

Debolezza: traffico analizzabile

Problema: i reply-block possono essere usati più di una volta (anche per inviare risposte diverse)

$\Rightarrow$ **Attacco:** chi può

1. monitorare il traffico di posta in ampie porzioni della rete e
2. inviare un numero enorme di risposte usando lo stesso reply-block

(*replay attack*), può ricostruire il percorso rilevando un improvviso e anomalo aumento del traffico tra gli elementi della catena

Contromisura: nessuna (con i remailer cypherpunk)

N.B.: sarebbe meglio se i reply-block fossero monouso

Tipo II

Remailer mixmaster

Remailer di **tipo II**:

- sono *ancora* più sicuri dei remailer cypherpunk
- non hanno basi dati di utenti, né registri (log)
- i messaggi anonimi sembrano tutti provenire dallo stesso utente (fittizio) del remailer
- i messaggi inviati all'utente fittizio rimbalzano
- richiedono un client apposito, lato utente
- si basano su un protocollo specifico (ma sempre implementato sopra SMTP)
- possono anche funzionare in modalità cypherpunk

Protocollo mixmaster (1)

- concatenazione e cifratura predisposti automaticamente dal client
- il messaggio può essere compresso (formato GZIP [RFC 1952])
- il messaggio è scomposto in *pacchetti mixmaster* di lunghezza fissa

  $\Rightarrow$ impossibili gli attacchi basati sulla lunghezza!
- ogni pacchetto è sottoposto a cifratura RSA + 3DES (non OpenPGP) e trasmesso via SMTP
- ogni remailer della catena conosce solo provenienza e destinazione (intermedie) del pacchetto

Protocollo mixmaster (2)

- solo l'ultimo remailer della catena vede quali pacchetti compongono un singolo messaggio
- reordering basato su un *pool dinamico temporizzato*
- rotazione automatica delle chiavi (annuale)
- ogni remailer conserva in una *replay cache* gli identificatori dei pacchetti elaborati di recente e si rifiuta di inviare una seconda volta un pacchetto già trasmesso

  $\Rightarrow$ impossibili i replay attack!

Protocollo mixmaster (3)

- traffico di copertura che nasconde i pacchetti significativi in mezzo ad un ‘rumore’ variabile: *dummy messages*
  - generati casualmente dai remailer
  - generati dai client
- meccanismo di gestione automatica degli abusi (opt-out)
- retrocompatibilità: i messaggi cypherpunk vengono trasmessi incapsulati nel protocollo mixmaster (quando possibile)

Vedi `http://www.obscura.com/~loki/remailer/remailer-essay.html`

Client mixmaster

http://mixmaster.sourceforge.net/

- licenza mixmaster (**libera**)
- interfaccia:
  - a riga di comando
  - a menu (ncurses)
- usabile da dentro un MUA (Mutt)

N.B.: insieme al client è distribuito anche il server (remailer)

Client mixmaster: update

- aggiornamento di
  - lista dei remailer mixmaster attivi
  - loro chiavi pubbliche
  - statistiche di affidabilità
- tre modalità:
  1. manuale
  2. periodico (cron)
  3. ad ogni connessione PPP (ip-up)
- il client ottiene questi dati da un *pinger*

Client mixmaster: menu (1)

Menu principale:

```
Mixmaster 3.0a9                Copyright Anonymizer Inc.


  0 outgoing messages in the pool.


          m)ail
          p)ost to Usenet
          r)ead mail (or news article)
          d)ummy message
          s)end messages from pool
          q)uit
```

Client mixmaster: menu (2)

Preparazione di un messaggio anonimo:

```
Mixmaster 3.0a9 - sending mail


r)edundancy:   1 copies

d)estination: dest@porto.it
s)ubject: Buone notizie

pgp encry)ption: no


m)ail message   e)dit message   f)ile   q)uit

c)hain: D,C,B,A    (reliability:  92.26%)
```

definizione della catena (a scelta oppure casuale)

Debolezza: cache limitata

Problema: per limitare la crescita della replay cache, gli identificatori vecchi vengono eliminati

$\Rightarrow$ **Attacco:** è possibile aspettare che il remailer si sia dimenticato di un vecchio pacchetto per inviarlo di nuovo: attacco basato su *analisi statistica* del traffico

Contromisura: nessuna (con i remailer mixmaster)

N.B.: sarebbe meglio se la replay cache venisse svuotata solo ad ogni rotazione della chiave del remailer

Debolezza: pinger incoerenti

Problema: il funzionamento dei pinger non è specificato dal protocollo mixmaster: i dati forniti possono essere incoerenti e non sincronizzati

$\Rightarrow$ **Attacco:** un pinger compromesso può fornire dati distorti ad alcuni client; un attaccante può sfruttare le differenze tra i dati ottenuti da client che hanno interrogato pinger diversi (*partitioning attack*)

Contromisura: nessuna (con i remailer mixmaster)

N.B.: sarebbe meglio se i dati forniti dai pinger fossero tutti coerenti e sincronizzati tra loro

Mixmaster: risposte?

- il protocollo mixmaster non supporta le risposte, né i destinatari anonimi
- si sfrutta la retrocompatibilità cypherpunk

  $\Rightarrow$ vulnerabilità ai replay attack

N.B.: sarebbe meglio se il protocollo supportasse direttamente dei reply-block monouso

Tipo III

Remailer mixminion

Remailer di **tipo III**:

- tipo sperimentale attualmente in sviluppo
- non hanno basi dati di utenti, né registri (log)
- i messaggi anonimi sembrano tutti provenire dallo stesso utente (fittizio) del remailer
- supportano direttamente le risposte
- richiedono un client apposito, lato utente
- si basano su un protocollo specifico (che sfrutta connessioni TLS sopra TCP)
- possono anche funzionare in modalità cypherpunk o mixmaster

Sviluppo di mixminion

L'implementazione di riferimento è giunta alla versione 0.0.7.1 (10 maggio 2004).

Licenza: (per ora) LGPL; (tra poco) Expat (MIT).

Home page:
`http://www.mixminion.net/`

- notizie
- documentazione
- programmi
- …

Protocollo mixminion (1)

- concatenazione e cifratura predisposti automaticamente dal client
- il messaggio è scomposto in *pacchetti mixminion* di lunghezza fissa
- ogni pacchetto è sottoposto a cifratura RSA + LIONESS (?) e trasmesso su connessione TLS

  $\Rightarrow$ cifratura con chiave di sessione effimera: impossibile anche per l'amministratore decifrare il traffico passato
- ogni remailer della catena conosce solo provenienza e destinazione (intermedie) del pacchetto

Protocollo mixminion (2)

- solo l'ultimo remailer della catena vede quali pacchetti compongono un singolo messaggio
- le risposte sfruttano reply-block monouso (SURB: *Single-Use Reply-Block*)
- sono possibili:
  1. messaggi anonimi (**forward**)
  2. risposte dirette (**direct reply**)
  3. risposte anonimizzate (**anonymized reply**)
- gli stessi remailer non possono distinguere i tre tipi di pacchetto
- reordering basato su un *pool dinamico temporizzato*

Protocollo mixminion (3)

- rotazione automatica delle chiavi (configurabile)
- ogni remailer conserva in una *replay cache* gli hash delle intestazioni elaborate dall'ultimo cambio di chiave e si rifiuta di inviare una seconda volta un pacchetto già trasmesso

  $\Rightarrow$ impossibili gli attacchi statistici!
- traffico di copertura casuale (tra remailer)
- meccanismo di gestione automatica degli abusi (opt-out con segreto)
- retrocompatibilità cypherpunk e mixmaster (per incapsulamento)

Vedi `http://www.mixminion.net/minion-design.pdf`

Debolezza: payload alterabile

Problema: risposte indistinguibili dai messaggi $\Rightarrow$ impossibili verifiche di integrità del payload

$\Rightarrow$ **Attacco:** chi può alterare una parte del payload (*tagging attack*), può riconoscere il pacchetto corrispondente in un punto successivo della catena perché la decifratura produce un risultato non conforme al formato atteso

Contromisura: verifica di integrità delle intestazioni e metodo dello scambio (*swap method*)

N.B.: il protocollo mixmaster è vulnerabile a questo tipo di attacco

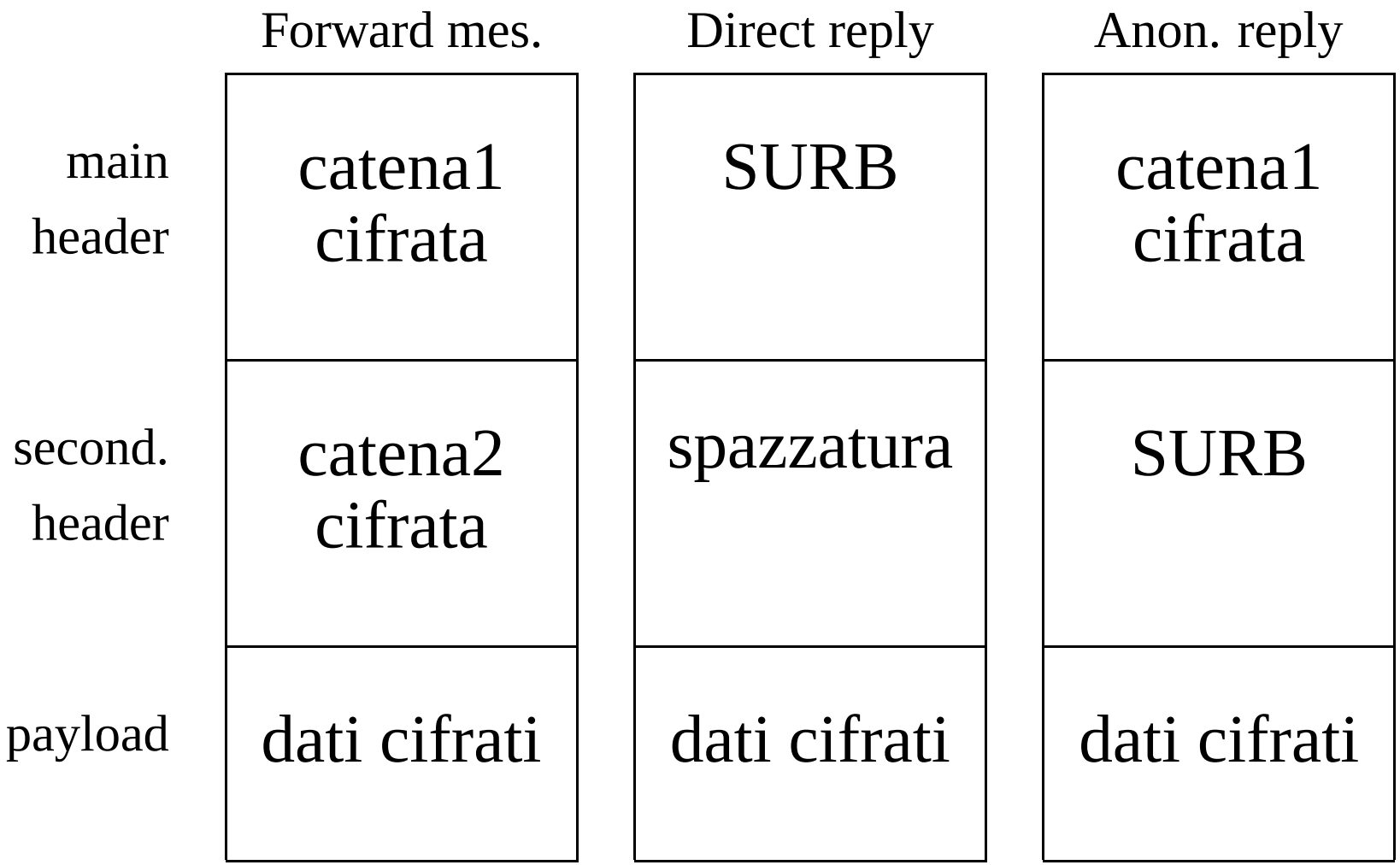
Mixminion: swap method (1)

- il pacchetto mixminion è composto da **due intestazioni** (principale e secondaria) e dal carico pagante
- la catena di remailer è suddivisa in **due "gambe"**
- ogni intestazione è suddivisa in sottointestazioni (una per ogni elemento della gamba)
- in ogni sottointestazione:
  1. hash del resto dell'intestazione per verificare l'integrità del percorso
  2. chiave simmetrica LIONESS per decifrare il resto del pacchetto
- l'intestazione secondaria è cifrata con LIONESS usando un **hash del payload** come chiave

Mixminion: swap method (2)

- il remailer alla fine della prima gamba (**crossover point**) effettua l'operazione di *scambio*:
  1. calcola la funzione hash del payload
  2. decifra l'intestazione secondaria
  3. scambia le due intestazioni
- la cifratura LIONESS ha le seguenti proprietà:
  - l'operazione di decifratura può essere usata come metodo di cifratura senza perdita di sicurezza
  - chi non conosce la chiave, non può riconoscere il risultato della decifratura di un blocco alterato, né prevedere l'effetto di un'alterazione

Mixminion: pacchetto



Efficacia dello swap method

Tagging attack: alterazione del payload

Forward message: *prima gamba* $\rightarrow$ intestazione secondaria irrecuperabile; *seconda gamba* $\rightarrow$ mittente ormai anonimo

Direct reply: *prima gamba* $\rightarrow$ decifratura del payload ad ogni passo usata come cifratura: solo il destinatario può scoprire se c'è stata alterazione

Anonymized reply: *prima gamba* $\rightarrow$ come forward message; *seconda gamba* $\rightarrow$ come prima gamba direct reply

Mixminion: directory server

- metodo di aggiornamento dei dati sui remailer (lista, chiavi, statistiche) specificato dal protocollo mixminion
- i client ottengono i dati da un insieme di *directory server* sincronizzati e ridondanti
- dati **completi** e **firmati** da ogni directory server

  $\Rightarrow$ impossibili i partitioning attack!

Debolezza: rete mutevole

Problema: al passare del tempo i remailer entrano o escono (anche provvisoriamente) dalla lista dei nodi attivi

$\Rightarrow$ **Attacco:** chi può ritardare i messaggi di `mitt` finché un qualche remailer non viene eliminato dalle liste dei directory server, può poi supporre che qualunque messaggio usi ancora quel remailer provenga probabilmente da `mitt` (*trickle attack*)

Contromisura: i client aggiornano spesso i dati sulla rete, ma aspettano un po' prima di usare i nuovi remailer; viene inoltre generato un traffico di copertura verso i vecchi remailer

Mixminion: nym server?

- i nym server basati sulla rete cypherpunk conservano una base dati di corrispondenze nym↔reply-block
- con mixminion i reply-block sono monouso
- approcci possibili per realizzare nym server basati su mixminion:
  1. il nym è associato ad una coda di SURB da usare uno per ogni messaggio in arrivo
     ⇒ vulnerabilità ai *flooding attack* (DoS)
  2. il server conserva i messaggi per un nym finché l'utente non fornisce i SURB necessari all'invio (analogo al protocollo POP)
     ⇒ il server dovrà cifrare i messaggi in coda con la chiave pubblica del nym

Conclusioni

Conclusioni

- l'evoluzione dei remailer anonimi sta creando strumenti sempre più efficaci nel consentire comunicazioni anonime
- altro uso: comunicazioni non osservabili (`mitt` e `dest` comunicano attraverso la rete dei remailer, ma firmano digitalmente i messaggi)
- purtroppo l'utilità è limitata da:
  1. scarsa diffusione di conoscenza
  2. scarso numero di remailer attivi
  3. scarso numero di utenti

Bibliografia (1)

Sulla privacy e l'anonimato in generale:

http://e-privacy.firenze.linux.it/

http://www.winstonsmith.info/

http://e-privacy.firenze.linux.it/2002/documenti/Kriptonite_libro.pdf

http://www.bigbrotherawards.org/

http://www.google-watch.org/

Sui remailer anonimi:

http://www.stayinvisible.com/index.pl/e_privacy_remailers

http://riot.eu.org/anon/intro.html.it

http://anon.efga.org/Remailers

Inviare un messaggio con subject "`remailer-help`" all'indirizzo di un remailer per ottenere una guida all'uso

Bibliografia (2)

Mappa dei remailer:

`http://riot.eu.org/anon/remap.html`

Statistiche di affidabilità dei remailer:

`http://www.noreply.org/remsaint.php`

Inviare un messaggio con subject “`remailer-key`” all’indirizzo di un remailer per ottenere le sue chiavi pubbliche

Sui nym server:

`http://lexx.shinn.net/nym/`

`http://riot.eu.org/anon/doc/nym.html`

Grazie dell’attenzione!