CARNET

Bitmessage: komunikacija koja ne otkriva metapodatke

CERT.hr-PUBDOC-2018-7-363

CERT.hr

Sadržaj

Ovaj dokument izradio je Laboratorij za sustave i signale Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

1 Uvod

U privatnim i u poslovnim okolinama, sve više komunikacije obavlja se putem računala i računalnih mreža. Sigurnost i privatnost digitalne komunikacije nije nova tema – građani i organizacije koji žele sigurno i privatno komunicirati već duže vrijeme koriste protokole i alate namijenjene za to. Danas su raširenи standardi *OpenPGP* i *S/MIME* za zaštitu elektroničke pošte te razni novi protokoli i aplikacije za privatnu komunikaciju, primjerice aplikacija *Signal* opisana u prethodnom dokumentu Nacionalnog CERT-a i *Off-the-Record Messaging* komunikacijski protokol. No, navedeni načini komunikacije i dalje imaju dva značajna problema.

Kao prvo, zbog problema razmjene ključa (u obliku *OpenPGP* javnih ključeva, *S/MIME* certifikata i sl.), ispravno korištenje takvih rješenja često nije praktično. Starija rješenja, primjerice *OpenPGP*, zahtijevaju od korisnika osnovno razumijevanje kriptografije kako bi ostvarili sigurnu komunikaciju. Neka od novih rješenja su praktičnija po ovom pitanju – aplikacija *Signal* primjerice dopušta korisnicima da provjere ispravnost ključeva pomoću QR koda, dok *Off-the-Record Messaging* protokol omogućava provjeru autentičnosti pomoću zajedničke tajne. No u konačnici, korisnici uvijek moraju na neki aktivan način provjeriti autentičnost komunikacije. Posljedica toga je da velik broj korisnika i dalje ili ne koristi ovakve protokole odnosno aplikacije za sigurnu komunikaciju ili, ako ih već koriste, onda ne provjeravaju autentičnost komunikacije.

Drugi, možda još i veći problem je zaštita tajnosti metapodataka. Kratko rečeno – metapodaci su podaci koji opisuju druge podatke. Primjerice, u elektroničkoj pošti, metapodaci su između ostaloga adresa pošiljatelja i primatelja poruke, naslov poruke i vrijeme slanja poruke. U komunikaciji aplikacijom *Signal*, metapodaci uključuju broj telefona pošiljatelja i primatelja. U gotovo svim komunikacijskim protokolima ti metapodaci nisu prikriveni, tj. iako napadači ne mogu otkriti sadržaj komunikacije, često mogu otkriti tko je s kim i kada komunicirao.

Na prvi pogled, ti podaci se možda i ne čine bitnima, no u stvarnosti prikrivanje metapodataka je ključan dio privatnosti komunikacije. Značaj metapodataka dobro ilustriraju sljedeća dva citata bivših dužnosnika obavještajnih agencija SAD-a:

„Metapodaci vam govore sve o nečijem životu. Ako imate dovoljno metapodataka, zapravo i ne trebate sadržaj.“

- Stewart Baker, bivši glavni pravni savjetnik NSA-a (1)

„Mi ubijamo ljude na temelju metapodataka.“

- General Michael Hayden, bivši voditelj NSA-a i CIA-a (1)

U idealnom slučaju težimo tome da metapodaci prilikom komunikacije budu prikriveni, no to je izrazito teško postići. Primjerice, kako će poslužitelj elektroničke pošte znati kome treba dostaviti poruku ako mu je upravo ta informacija uskraćena?

Bitmessage je protokol napravljen upravo kao rješenje navedenih problema. *Bitmessage* omogućava korisnicima sigurnu komunikaciju i prikrivanje svih metapodatka, a da pri tome korisnik ne mora dodatno provjeravati autentičnosti ključeva. *Bitmessage* je često

predstavljen kao decentralizirana i sigurna alternativa elektroničkoj pošti. U ovom dokumentu bit će objašnjena osnovna načela rada *Bitmessage* protokola te neke od njegovih naprednih značajki.

2 Osnovno o *Bitmessageu*

Bitmessage kao komunikacijski protokol značajnim je dijelom inspiriran načelima rada kriptovalute *Bitcoin*. Glavna sličnost s kriptovalutom *Bitcoin,* i jedno od temeljnih svojstva *Bitmessage* protokola, je to da je on decentraliziran. Drugim riječima, rad protokola ne ovisi o nekom središnjem poslužitelju, već se prilikom rada poruke šalju kroz mrežu korisnika. Preciznije, korisnici *Bitmessagea* tvore takozvanu *peer-to-peer* (P2P) mrežu, sličnu mreži kriptovalute *Bitcoin* i protokola za dijeljenje datoteka *BitTorrent*.

Slanje i primanje poruke *Bitmessage* protokolom bit će pojednostavljeno objašnjeno kroz primjer. U sljedećem primjeru, korisnik Ante želi putem *Bitmessage* protokola poslati poruku korisniku Borni:

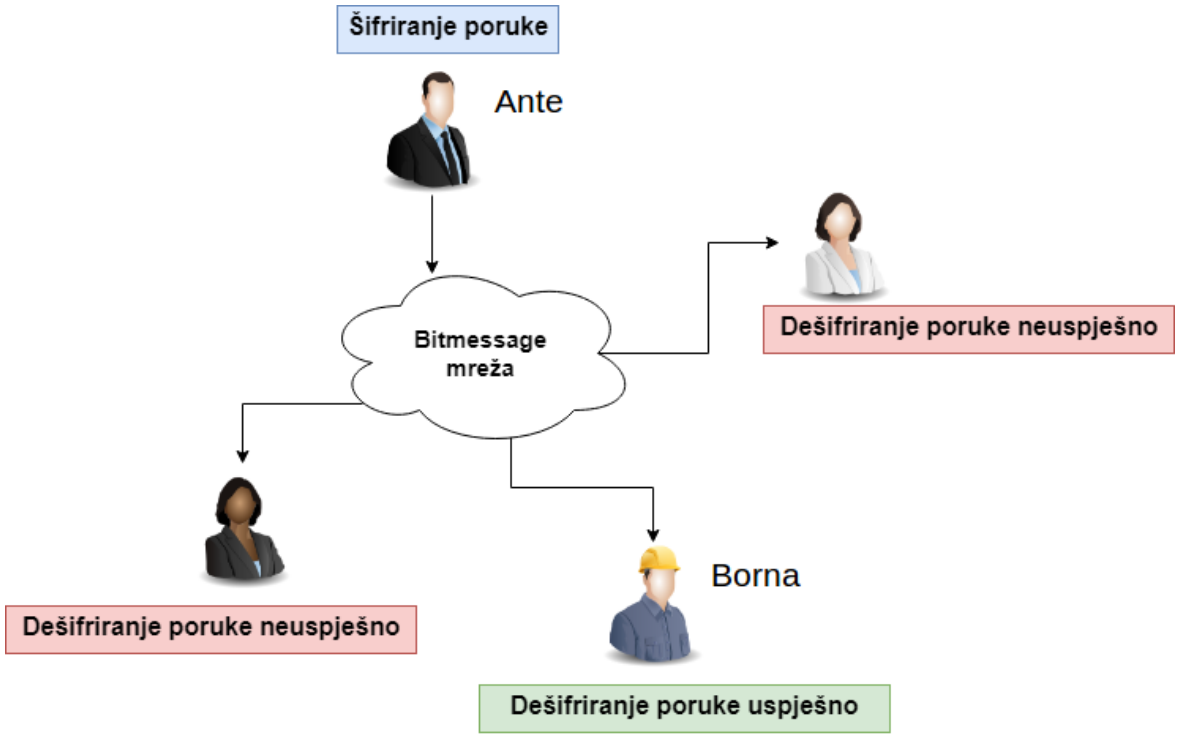
1. Prvo, Ante šifrira poruku tako da ju može dešifrirati samo Borna. Kriptografskim rječnikom, Ante šifrira poruku pomoću Borninog javnog ključa.

2. Zatim, Ante šalje tu šifriranu poruku svim korisnicima *Bitmessage* mreže.

3. Svaki korisnik zatim pokuša dešifrirati poruku. Ako ne uspije dešifrirati poruku, odbacuje ju – to znači da poruka nije namijenjena njemu.

4. Ako korisnik uspješno dešifrira poruku, to znači da je poruka namijenjena baš njemu te mu se ona u konačnici i prikazuje. U ovom slučaju, samo će Borna moći dešifrirati i pročitati poruku. S kriptografske strane – Borna je jedini koji može dešifrirati poruku jer on jedini posjeduje odgovarajući privatni ključ (koji pripada javnom ključu kojim je poruka šifrirana).

Na slici 1 prikazan je dijagram ovog pojednostavljeno objašnjenog procesa slanja i primanja poruke.



Slika 1 – Ante šalje Borni poruku protokolom *Bitmessage*

Ovakav način slanja poruka omogućava da su i sadržaj poruke i njeni metapodaci šifrirani. Kako se *Bitmessage* poruka šalje svima, adrese pošiljatelja i primatelja zapravo i nisu potrebne za prijenos poruke. Napadači koji prisluškuju mrežni promet vide samo šifrirane poruke koje putuju cijelom mrežom – ne vide nikakve adrese, brojeve telefona i slično.

S tehničke strane, ovakav pristup razmjeni poruka se možda čini rastrošnim i neučinkovitim u komunikacijskom smislu. Naime, ako svatko šalje svima poruke, kako bi ovakav sustav uopće mogao funkcionirati s velikim brojem korisnika i poruka? Za ovaj problem skaliranja osmišljeno je zanimljivo rješenje koje će biti opisano u sljedećem poglavlju.

Ovime je objašnjeno kako se poruke razmjenjuju i šifriraju. No očigledno je da prije šifriranja treba znati javni ključ primatelja. Dakle, treba na neki siguran način razmijeniti ključeve. Konkretnije, u korištenom primjeru – kako bi Ante mogao šifrirati poruku za Bornu, on prvo mora na siguran način saznati njegov javni ključ. U uvodu je navedeno kako za razliku od *OpenPGP*-a, *Signala* i sličnih rješenja, korisnici *Bitmessagea* ne moraju provoditi dodatne korake kako bi razmijenili ključeve te provjerili autentičnost sugovornika. Ključ tog mehanizma su *Bitmessage* adrese. *Bitmessage* adrese oblika su **BM-*<niz od 30-ak znakova>***, primjerice:

BM-BcbRqcFFSQUUmXFKsPJgVQPSiFA3Xash

Bitmessage adrese inspirirane su *Bitcoin* adresama – navedeni niz od 30-ak znakova **sadrži kriptografski sažetak (eng. *cryptographic hash*) javnog ključa** koji odgovara toj adresi.

U svrhu sigurne razmjene ključa, kada se koristi *OpenPGP* standard i slična rješenja, pošiljatelj prvo mora saznati adresu primatelja, zatim mora nekako doći do javnog ključa primatelja i konačno mora provjeriti odgovara li zaista taj javni ključ primatelju. U *Bitmessage* protokolu taj proces je drugačiji – na primjeru Ante koji šalje poruku Borni, prije slanja poruke Ante obavlja sljedeće:

1. Ante na neki način sazna da je Bornina Bitmessage adresa *BM-BcbRqcFFSQUUmXFKsPJgVQPSiFA3Xash*. Primjerice, Borna mu uživo kaže adresu, Ante ju pronađe na Borninoj Web stranici ili slično.

2. Zatim, Ante kroz *Bitmessage* mrežu svim korisnicima šalje zahtjev za javnim ključem povezanim s pronađenom *Bitmessage* adresom.

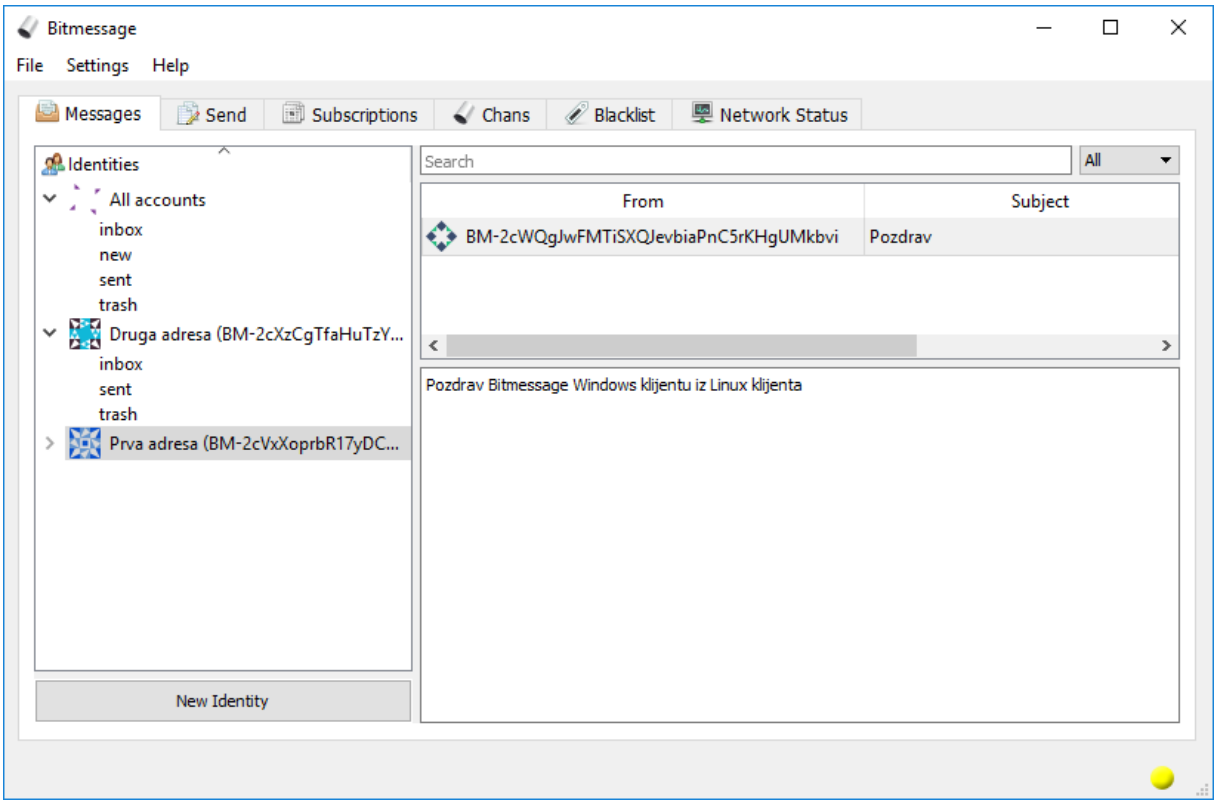
3. Kao odgovor na taj zahtjev, Borna šalje svim korisnicima poruku koja sadrži njegov javni ključ uz još neke dodatne informacije.

4. Konačno, Ante prima javni ključ, računa njegov kriptografski sažetak pa ga uspoređuje sa sažetkom iz *Bitmessage* adrese. Ako su jednaki, može biti siguran da je dobio javni ključ koji odgovara adresi iz prvog koraka.

Za krajnjeg korisnika, ovaj proces je značajno jednostavniji od korištenja *OpenPGP*-a i sličnih rješenja – Ante mora samo saznati adresu, dok korake 2, 3 i 4 automatski obavlja njegov *Bitmesage* klijent (program za komuniciranje *Bitmessage* protokolom). Anti ne treba znati kriptografiju niti razumijeti ovaj proces. Sve što on treba znati je da mora saznati adresu sugovornika – o svemu ostalome brine *Bitmessage*.

Jedan očigledni nedostatak *Bitmessage* adresa njihova je složenost. Jer, puno je lakše zapamtiti adresu e-pošte oblika *ime.prezime@domena* nego niz nasumičnih znakova s prefiksom „*BM-*“. No, nije ispravno uspoređivati *Bitmessage* adrese s adresama e-pošte jer adrese e-pošte same po sebi nisu dovoljne za sigurnu komunikaciju. Kada bi za usporedbu gledali e-poštu zaštićenu *OpenPGP* standardom, razmjenom *Bitmessage* adrese u isto vrijeme se dobiva i adresa e-pošte i sažetak javnog ključa. Iz te perspektive, lako je vidjeti prednost korištenja ovakvog oblika adresa – ako korisnik ima nečiju adresu, onda istovremeno ima i sve što mu je potrebno za sigurnu komunikaciju.

Do sada je u tekstu opisivan *Bitmessage* protokol – računalni jezik kojim „razgovaraju“ računala u *Bitmessage* mreži. Kako bi krajnji korisnici zapravo koristili taj protokol, oni koriste neki od *Bitmessage* klijenata – programa za komunikaciju *Bitmessage* protokolom. Službeni *Bitmessage* klijent naziva se *PyBitmessage*, a dostupan je za operacijske sustave Windows, Mac i Linux. Poveznice na najnoviju inačicu dostupne su ovdje. U trenutku pisanja ovog dokumenta najnovija inačica bila je 0.6.3.2, izdana 13. veljače 2018. godine te je ona i prikazana u ovom dokumentu. Na slici 3 prikazano je sučelje *PyBitmessage* klijenta s primjerom primljene poruke.



Slika 2 – sučelje *PyBitmessagea*, službenog *Bitmessage* klijenta

3 Napredne značajke i koncepti *Bitmessagea*

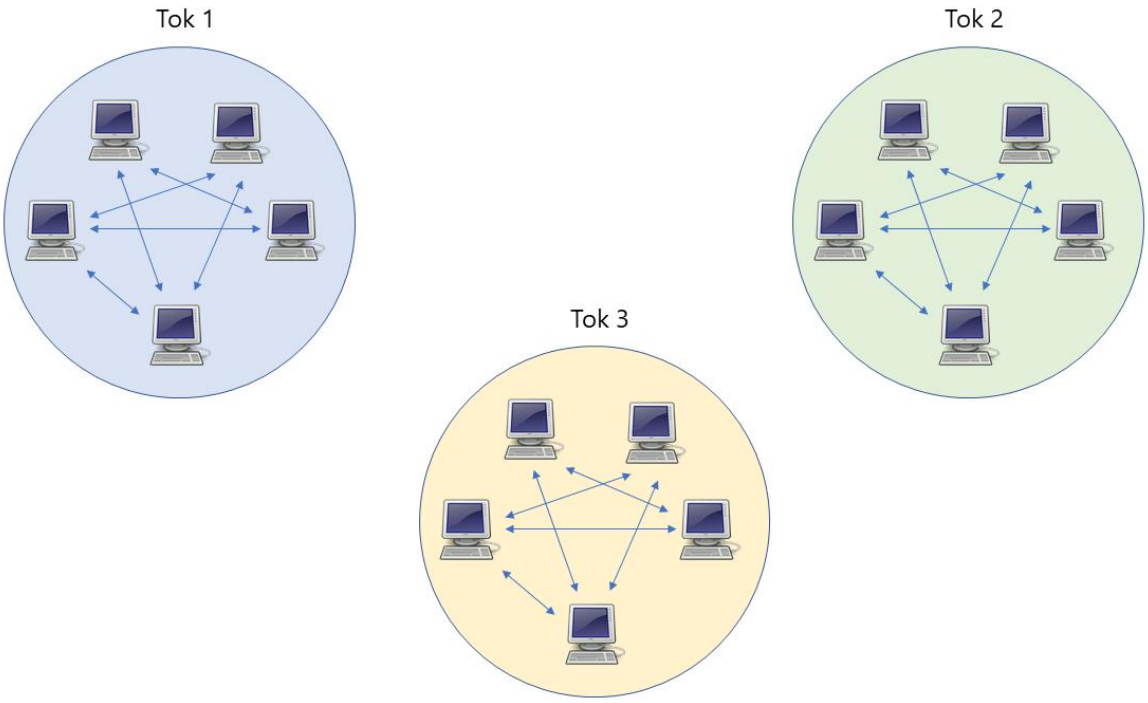
Rad *Bitmessage* protokola temelji se i na nekim naprednim konceptima te, osim osnovne komunikacije između dva krajnja korisnika, podržava i niz naprednih funkcionalnosti od kojih će neke biti objašnjene u ovom poglavlju.

3.1 Tokovi (eng. *streams*)

Kako bi se riješio problem skalabilnosti, tj. kako bi *Bitmessage* mogao funkcionirati s velikim brojem korisnika i poruka, u specifikaciji *Bitmessage* protokola predviđeno je odvajanje korisnika, tj. njihovih pojedinih adresa, u takozvane tokove (eng. *streams*). Tokovi su grupe adresa – oni su zapravo kao manje, odvojene *Bitmessage* mreže. Svaka adresa dodijeljena je jednom i samo jednom toku. Odvajanjem u tokove, ne bi više svaki korisnik *Bitmessagea* svakome drugome slao poruke, već bi se koncept „svatko šalje svakome“ primjenjivao samo unutar toka.

U daljnjem tekstu pretpostavit će se da svaki korisnik ima točno jednu *Bitmessage* adresu te se zato i nalazi u točno jednom toku. Ta pojednostavljena situacija dovoljna je za objašnjavanje principa tokova. U slučaju kada korisnik ima više adresa u više različitih tokova, temeljna načela rada se gotovo uopće ne mijenjaju.

Koncept odvajanja korisnika (odnosno njihovih adresa) u tokove te razmjene poruka unutar tokova ilustriran je na slici 3.

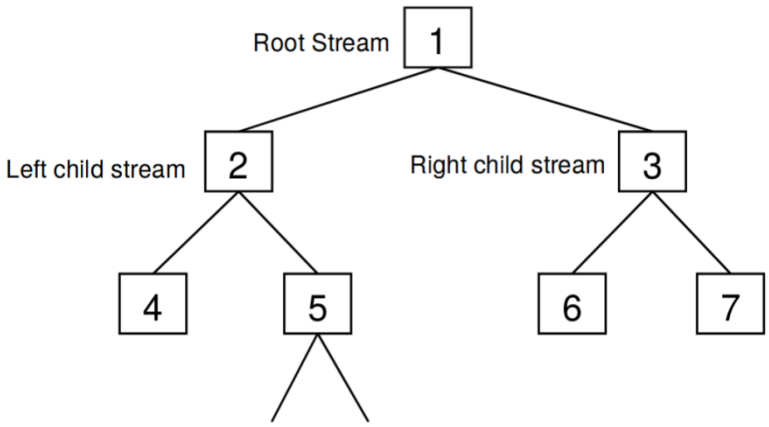


Slika 3 – odvajanje korisnika (tj. njihovih adresa) u tokove i razmjena poruka unutar toka

No, postavlja se pitanje – kako onda korisnici čije se adrese nalaze u različitim tokovima mogu komunicirati? Najjasnije je objasniti na primjeru – recimo da Ante iz toka 4 želi

poslati poruku Borni u toku 7. Da Ante pošalje poruku Borni, dovoljno je da tu poruku pošalje nekolicini korisnika iz toka 7 – ti korisnici će zatim zbog principa rada *peer-to-peer* (P2P) mreže proširiti tu poruku kroz ostatak toka 7, te će u konačnici poruka stići i do Borne. No, problem je u tome što Ante ne zna ni jednog korisnika iz toka 7.

Kako bi korisnik iz jednog toka (u primjeru – Ante) saznao informacije o nekolicini korisnika iz drugog toka (u primjeru – toka 7), predviđen je sustav temeljen na hijerarhiji tokova u obliku binarnog stabla (2), kao što je prikazano na slici 4.



Slika 4 – hijerarhija tokova u obliku binarnog stabla (izvor)

Kao preduvjet, da bi predviđeni sustav funkcionirao, korisnici trebaju:

- održavati kratki popis od nekoliko korisnika iz prvog (korijenskog) toka,
- povremeno oglašavati svoje postojanje u nadređenom toku,
- te održavati popis korisnika iz svojeg toka i iz izravno podređenih tokova.

Broj toka kojem pripada korisnikova adresa sadržan je u samoj *Bitmessage* adresi. Objašnjeno kroz primjer Ante iz toka 4 koji želi poslati poruku Borni iz toka 7, cijeli proces slanja poruke funkcionira na sljedeći način:

1. Prvo, Ante iz Bornine *Bitmessage* adrese pročita informaciju da se Borna, odnosno ta Bornina adresa, nalazi u toku 7. Ante sada zna da poruku za Bornu mora proširiti kroz tok 7. To može učiniti slanjem te poruke nekolicini korisnika u toku 7, koji će zatim poruku proširiti kroz ostatak toka. No, problem je u tome što Ante ne zna ni jednog korisnika iz toka 7.

2. Kao što je prikazano na slici 2, zbog strukture binarnog stabla, Ante zna da je tok 7 podređen toku 3 te da je taj tok podređen toku 1. Zato Ante svoju potragu za korisnicima iz toka 7 započinje u korijenskom toku, toku 1. Tamo on pita nekolicinu korisnika da mu pošalju popis nekoliko korisnika koje oni poznaju iz toka 3.

3. Sada, kada Ante zna nekoliko korisnika iz toka 3, on njih pita za popis nekoliko korisnika iz toka 7.

4. Zatim, kada Ante zna nekoliko korisnika iz toka 7, on njima šalje poruku te ih traži da prošire poruku kroz ostatak njihovog toka.

5. Konačno, ti korisnici toka 7 proširuju poruku kroz cijeli tok te ju u nekom trenutku prima i Borna.

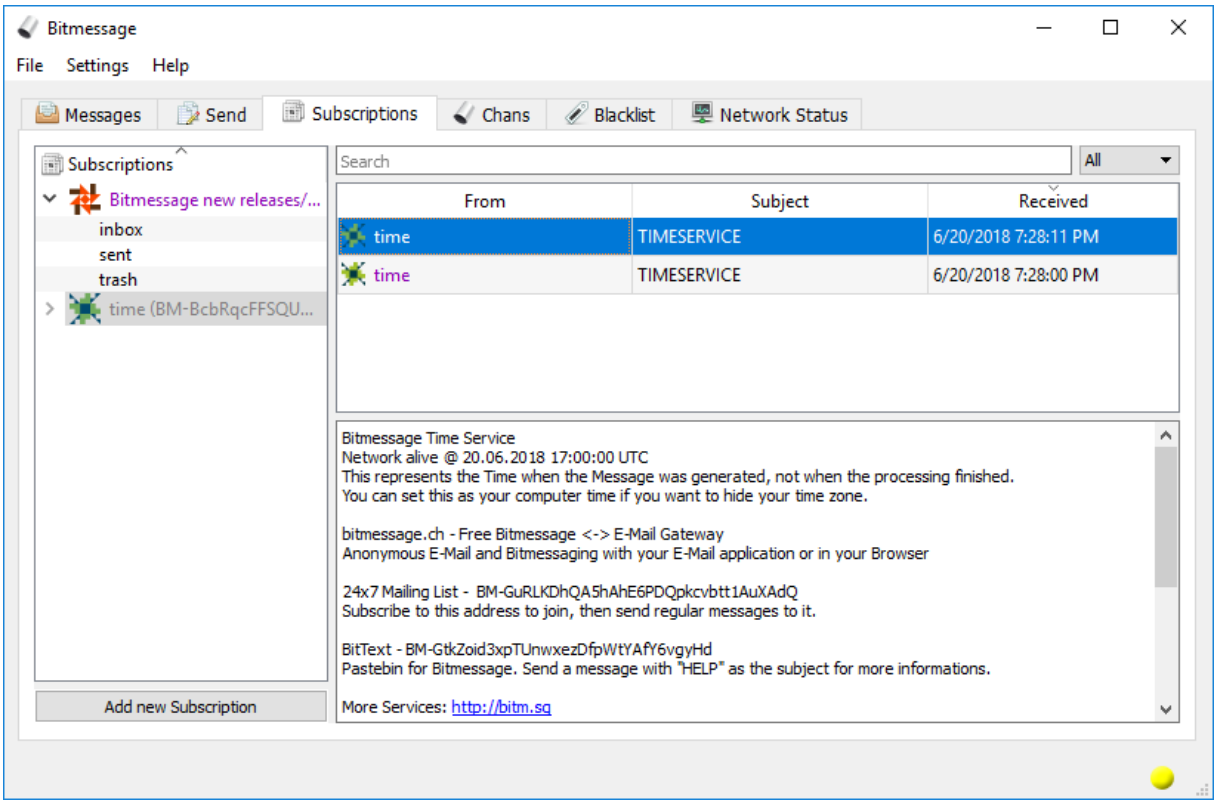
Ovakvim hijerarhijskim pristupom se potencijalno ogroman prostor „pretraži" s malim brojem poruka u vrlo kratko vrijeme. Trenutno, ovo rješenje je tek teoretski prijedlog jer je broj *Bitmessage* korisnika i dalje dovoljno nizak da se svi nalaze u istom toku te svi međusobno razmjenjuju poruke (3) (4).

3.2 Objava (eng. *broadcast*)

Uz slanje izravnih poruka jednom, točno određenom primatelju, neki korisnici imaju potrebu slati poruke i većem broju korisnika. Primjerice, osim što Ante, korisnik *Bitmessagea*, šalje pojedinačne, izravne poruke prijateljima, kolegama i obitelji on se i bavi istraživanjem novih tehnika za zaštitu privatnosti, pa povremeno želi poslati poruku grupi stručnjaka ili zainteresiranoj publici s novostima o tome što je otkrio. Kada ne bi koristio *Bitmessage*, Ante bi mogao slati bilten elektroničke pošte (eng. *newsletter*) na koji bi se zainteresirana publika pretplatila ili bi primjerice mogao objaviti taj sadržaj na svojemu blogu kojega bi zainteresirana publika pratila.

No, Ante to može postići i pomoću *Bitmessagea*. Unutar *Bitmessagea*, moguće je sa svoje adrese poslati i tzv. **objave (eng. *broadcasts*)** – poruke namijenjene većem broju korisnika. *Bitmessage* objave mogu se koristiti kao ekvivalent biltena (eng. *newsletter*) u elektroničkoj pošti ili sličnih mehanizama u drugim medijima komunikacije.

Za razliku od izravnih poruka, objave su posebne poruke koje primaju svi pretplatnici (eng. *subscribers*) neke adrese. Taj koncept bit će objašnjen na primjeru – kako bi Ante slao objave, prvi korak mu je razmijeniti svoju *Bitmessage* adresu s budućim pretplatnicima. Primjerice, on može objaviti adresu na svojoj Web stranici, na društvenim mrežama, na forumu i slično. Krajnji korisnici koji žele primati obavijesti (pretplatnici) od Ante trebaju samo unijeti Antinu adresu u sučelje za pretplate u svom *Bitmessage* klijentu, i od tada će moći primati sve njegove buduće objave. Na slici 5 prikazan je primjer primanja objave u *PyBitmessage* klijentu.



Slika 5 – primjer primanja objave u *PyBitmessage* klijentu

Kako se *Bitmessage* poruke u svakom slučaju šalju svim korisnicima, implementacija objava je prilično jednostavna. Za razliku od izravnih poruka koje su šifrirane javnim ključem primatelja, poruke objave šifrirane su ključem izvedenim iz *Bitmessage* adrese pošiljatelja. Na taj način, svaki korisnik koji zna adresu pošiljatelja može, ako želi, dešifrirati i pročitati njegove poslane objave. Taj koncept prikazan je na slici 6.

Slika 6 – implementacija slanja *Bitmessage* objava

3.3 Lista razašiljanja (eng. *mailing list*)

Grupna komunikacija neizostavna je funkcionalnost bilo koje tehnologije za digitalnu komunikaciju. Postoji jasna potreba za mehanizmom pomoću kojega bi primjerice grupa prijatelja mogla međusobno dogovoriti druženje, pomoću koje bi zaposlenici neke tvrtke mogli komunicirati kako bi zajednički obavili posao i slično.

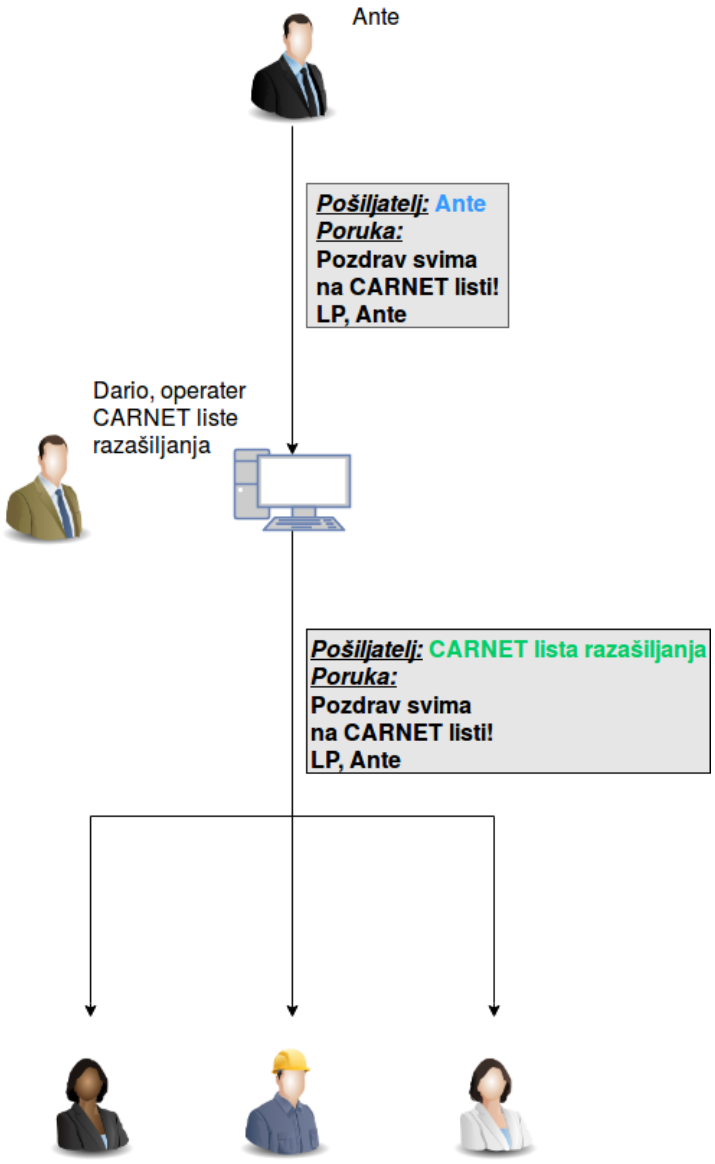
Za razliku od objava, kod kojih jedan korisnik šalje poruku mnogima, lista razašiljanja služi tome da bilo koji član liste može poslati poruku svim članovima liste.

U *Bitmessageu*, jedan mehanizam grupne komunikacije su *Bitmessage* liste razašiljanja (eng. *mailing lists*). One su slične istoimenom konceptu iz elektroničke pošte – slično kao u elektroničkoj pošti, moguće je napraviti *Bitmessage* adresu koja sve primljene poruke razašalje svojim pretplatnicima. Isto kao s objavama, za pretplatu na *Bitmessage* listu razašiljanja korisnik samo treba saznati njenu adresu te ju unijeti u odgovarajući dio sučelja klijenta.

S tehničke strane, *Bitmessage* lista razašiljanja jednostavno je implementirana kao adresa koja svaku primljenu poruku ponovno pošalje kao *Bitmessage* objavu (eng. *broadcast*). Jasnije je objasniti ovaj koncept na primjeru – recimo da Dario želi napraviti *Bitmessage* listu razašiljanja za zaposlenike CARNET-a kako bi oni mogli lakše zajednički komunicirati. Tada će on u svom *Bitmessage* klijentu napraviti novu listu razašiljanja, razmijenit će njenu adresu sa zaposlenicima CARNET-a, a oni će se u svojem klijentu na nju pretplatiti. Zatim, pretpostavimo da Ante želi poslati poruku svima na listi:

1. Prvo će Ante poslati poruku na adresu liste.
2. Zatim će Darijev *Bitmessage* klijent primiti tu poruku i poslati ju kao objavu (jer je Dario operater liste).
3. I, konačno, svi pretplatnici liste primit će tu objavu.

Ovaj primjer prikazan je na slici 7.



Slika 7 – primjer rada *Bitmessage* liste razašiljanja

Glavni nedostatak ovakvih lista je to što, za njihov neprekidan rad, operater liste (korisnik koji kontrolira njenu adresu) mora bez prestanka biti spojen na *Bitmessage* mrežu. Drugim riječima, ako operater liste razašiljanja nije spojen, lista razašiljanja neće raditi te poruke jednostavno neće biti razaslane.

3.4 Deterministično generiranje *Bitmessage* adrese

Uobičajeni način stvaranja *Bitmessage* adresa temelji se na nasumičnom generiranju kriptografskih ključeva. Rezultat takvog procesa je nasumična *Bitmessage* adresa te se taj postupak smatra prilično sigurnim. S negativne strane, kako bi korisnik pristupio porukama poslanima na tako generiranu *Bitmessage* adresu, on mora imati datoteku s odgovarajućim ključevima. To nije problem ako korisnik koristi *Bitmessage* na samo jednom uređaju – tada on ne treba ni znati da postoje ključevi koji su zapisani u neku datoteku. No, ako korisnik želi koristiti istu adresu na drugom uređaju, on mora prenijeti

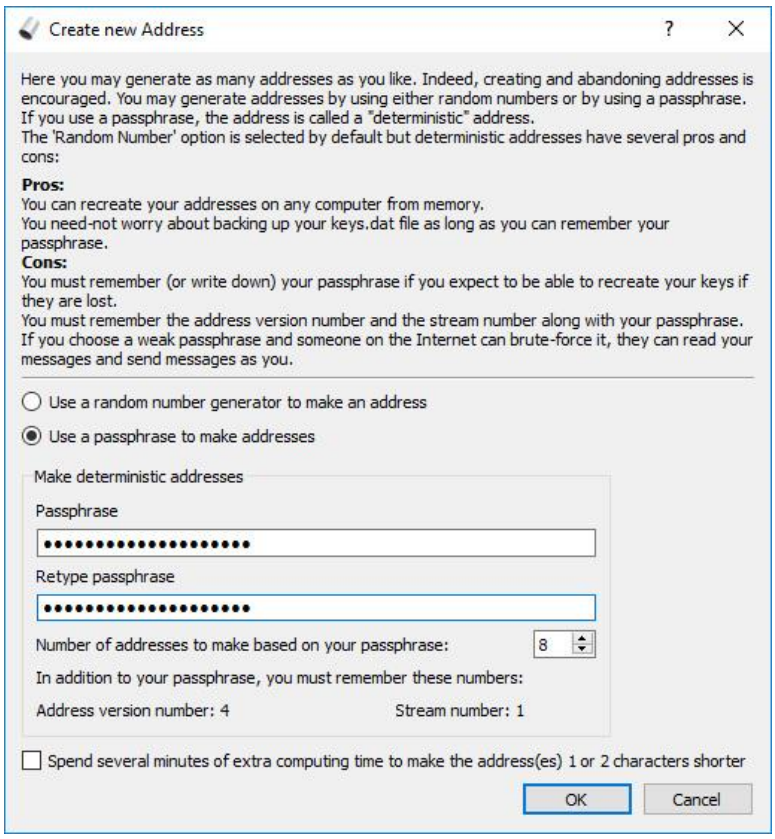
datoteku s ključevima na taj uređaj, a ako korisnik izgubi datoteku s ključevima, zauvijek će izgubiti i pristup svojoj *Bitmessage* adresi.

Upravo zato je kao alternativa nasumičnom generiranju adresa osmišljen i deterministički postupak generiranja. U ovom postupku korisnik prvo smisli sigurnu lozinku, te se zatim pomoću nje generiraju ključevi i odgovarajuća adresa. Nakon generiranja, ovi ključevi i adresa se nikako ne razlikuju od uobičajenih, nasumično generiranih ključeva i adresa – jedina razlika je način na koji su stvoreni.

Prednost ovog načina stvaranja adrese je to što je pri korištenju drugog uređaja dovoljno samo znati lozinku za korištenje iste adrese. Na drugom uređaju bit će dovoljno upisati istu lozinku, pa će se zatim na temelju nje generirati isti ključevi i adresa. Sigurnost ovog pristupa ovisi o sigurnosti lozinke – ako napadač može pogoditi lozinku koju je korisnik osmislio, onda može i generirati iste ključeve te u konačnici pristupiti porukama na generiranoj adresi.

Iako s deterističkim adresama ne treba voditi računa o stvaranju sigurnosnih kopija datoteka s ključevima, za njih je ključno dobro zapamtiti (ili na sigurno mjesto zapisati) lozinku, jer se gubljenjem lozinke trajno gubi mogućnost pristupa odgovarajućoj adresi.

Prilikom deterističkog generiranja adrese, moguće je zapravo stvoriti proizvoljan broj adresa iz iste lozinke. Do sada objašnjeni principi su u potpunosti isti za jednu ili za više adresa stvorenih iz lozinke. Na slici 6 prikazano je sučelje za stvaranje deterističkih adresa u *PyBitmessage* klijentu.



Slika 8 – sučelje za stvaranje deterističke adrese u *PyBitmessage* klijentu

3.5 Decentralizirane liste razašiljanja (eng. *decentralized mailing list*)

Deterministički način generiranja adresa omogućio je i jednu novu funkcionalnost – tzv. decentralizirane liste razašiljanja (eng. *decentralized mailing list*) ili, drugim nazivom, kanali (eng. *channel* ili skraćeno *chan*). Decentralizirane liste razašiljanja uvedene su zbog nedostataka običnih *Bitmessage* lista razašiljanja – obične liste razašiljanja su centralizirane jer iza njih stoji operater liste koji uvijek mora biti spojen na *Bitmessage* mrežu kako bi lista funkcionirala. Decentralizirane liste razašiljanja, kao što i ime kaže, imaju istu svrhu kao i obične liste razašiljanja. No, one su decentralizirane, a posljedica toga je da one neprekidno funkcioniraju te ne ovise ni o jednom pojedincu.

Bitmessage decentralizirane liste razašiljanja implementirane su kao deterministički generirane adrese čiju lozinku znaju svi članovi. Kako svi članovi znaju lozinku, tako mogu i generirati odgovarajuće ključeve te primati poruke za tu adresu. Na taj način, decentralizirane liste razašiljanja ne ovise ni o kakvim operaterima. Kada bismo ih uspoređivali s konceptima iz e-pošte, decentralizirane liste razašiljanja bile bi slične zajedničkom poštanskom sandučiću kojemu pristupaju svi članovi liste pomoću lozinke.

Primjerice, recimo da Dario ovaj puta želi napraviti decentraliziranu listu razašiljanja za zaposlenike CARNET-a:

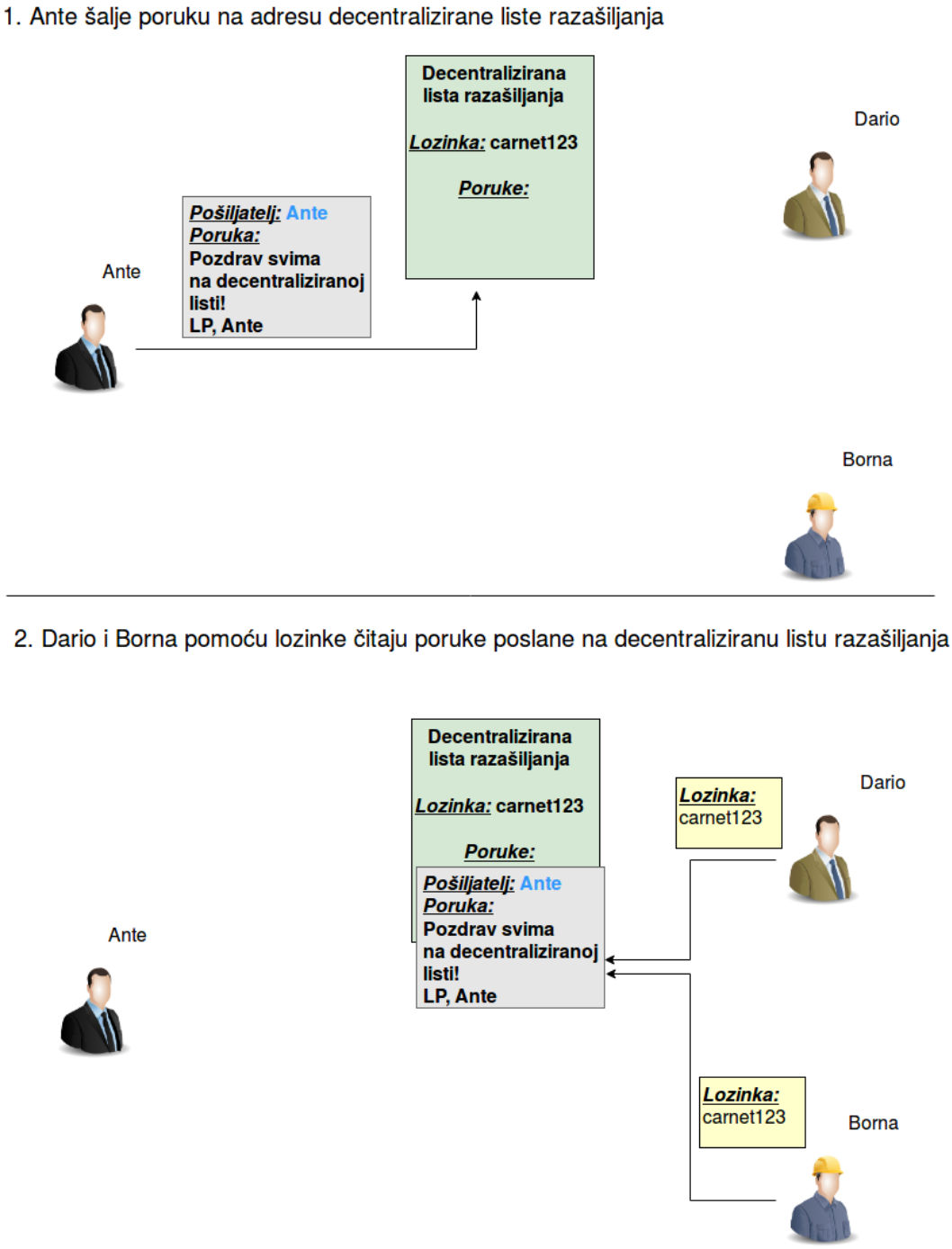
1. Dario će u svom *Bitmessage* klijentu odabrati stvaranje decentralizirane liste razašiljanja, smislit će i upisati lozinku liste te će pomoću te lozinke lista biti stvorena. Ključevi i adresa liste generiraju se deterministčkim postupkom iz upisane lozinke, kao što je objašnjeno u prethodnom poglavlju.

2. Zatim, Dario će zaposlenicima CARNET-a reći lozinku liste. Ako sadržaj poruka poslanih na listu treba biti tajan, ključno je osmisliti složenu lozinku te ju Dario i svi ostali članovi liste moraju držati tajnom.

3. Sada je Ante saznao lozinku od Darija i želi se pridružiti listi. To može učiniti upisivanjem lozinke u odgovarajući dio sučelja *Bitmessage* klijenta. Tada će se i kod njega generirati odgovarajući ključevi pa će i on moći pristupiti porukama poslanima na adresu liste.

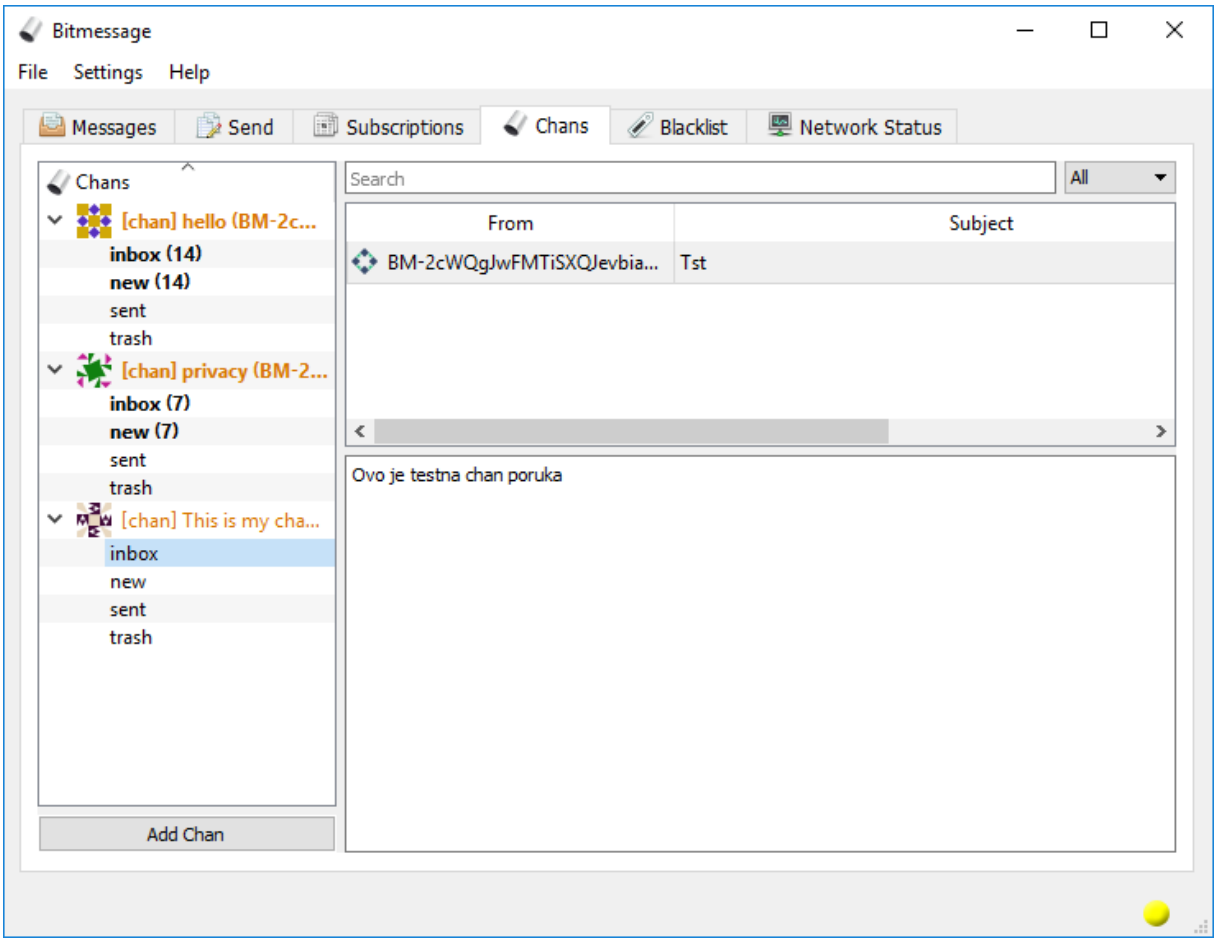
Bilo tko tko zna adresu liste može poslati poruku na njenu adresu, tako da ako je to nepoželjno, potrebno je i adresu liste držati tajnom. No samo korisnici s lozinkom, tj. s odgovarajućim ključevima, mogu čitati poruke poslane na adresu liste.

Koncept komunikacije kroz decentraliziranu listu razašiljanja prikazan je kroz primjer na slici 9. Na slici je prikazano kako Ante prvo šalje poruku na adresu decentralizirane liste razašiljanja, pa zatim Borna i Dario pomoću lozinke čitaju poruku poslanu na tu adresu.



Slika 9 – primjer slanja i čitanja poruke na decentraliziranoj listi razašiljanja

Na slici 10 prikazano je sučelje decentraliziranih lista razašiljanja (skraćeno nazvane *chans*) u *PyBitmessage* klijentu.



Slika 10 – sučelje decentraliziranih lista razašiljanja (*chans*) u *PyBitmessage* klijentu

3.6 Dokaz radom (eng. *proof-of-work)*

Jedan od velikih problema elektroničke pošte i drugih sličnih medija komunikacije su neželjene poruke (eng. *spam*, *junk mail* i sl.). Kada bi svatko mogao bez ograničenja slati poruke u *Bitmessage* mreži, to bi bio izrazito velik problem – ne samo da bi korisnici primali neželjene poruke, već bi i sama *Bitmessage* mreža vrlo brzo bila zagušena.

Rješenje tog problema u *Bitmessage* mreži, i još jedna sličnost s kriptovalutom *Bitcoin,* je tzv. dokaz radom (eng. *proof-of-work*). Kako bi poslao ispravnu poruku, pošiljateljev *Bitmessage* klijent mora prvo obaviti niz matematičkih operacija koje zahtijevaju određenu količinu računalnih resursa (utrošak vremena). Rezultat tih matematičkih operacija dodaje se poruci na način da primateljima služi kao dokaz da je pošiljatelj uložio navedenu količinu resursa. Prilikom primitka bilo koje poruke, primatelji prvo provjeravaju navedeni dokaz te prihvaćaju poruku isključivo ako dokaz smatraju zadovoljavajućim. Postupak generiranja i provjere dokaza osmišljen je tako da generiranje dokaza traje proizvoljno dugo (ovisno o parametru), a da provjera traje izrazito kratko. Ovakav način usporava vrijeme potrebno za slanje ispravne poruke, no služi važnoj svrsi – sprječava potencijalne napadače u slanju velikog broja neželjenih poruka.

Točna količina vremena odnosno resursa koju pošiljatelj mora uložiti u slanje poruke ovisi o više faktora. Jedan od faktora je vrijeme isteka poruke (eng. *time-to-live*, skraćeno *TTL*). Pojednostavljeno, to je vrijeme koje određuje koliko se dugo poruka zadržava u *Bitmessage* mreži (kako se poruke zadržavaju neko vrijeme u *peer-to-peer* mrežama, je izvan opsega ovog dokumenta). Primjerice, poruku s vremenom isteka dva dana primatelj može iz mreže preuzeti unutar dva dana od slanja. Nakon tog vremena poruka se briše iz mreže te, ako ju primatelj nije stigao preuzeti, pošiljatelj ponovno generira dokaz radom i šalje poruku. Pošiljatelj može proizvoljno odabrati vrijeme isteka poruke, no proporcionalno vremenu isteka poruke povećava se i količina resursa potrebnih za dokaz radom.

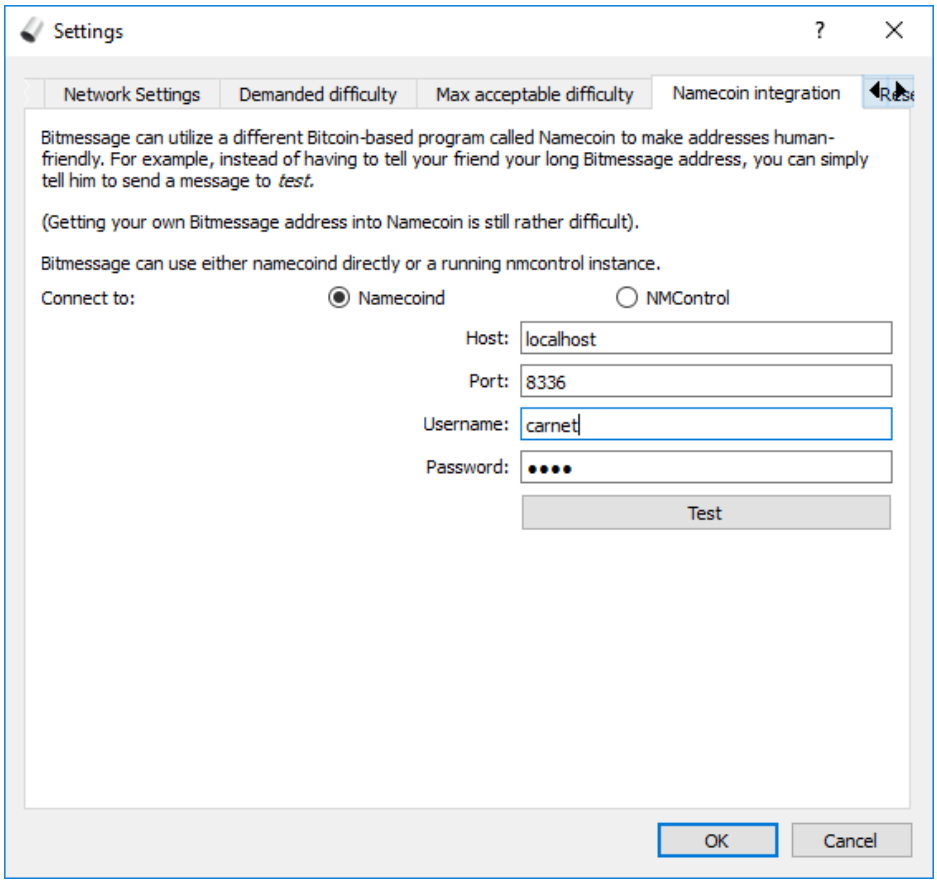
Osim vremena isteka poruke, resursi potrebni za dokaz radom ovise i o faktoru težine kojega bira primatelj za svoje dolazne poruke. Svaki korisnik može birati koliko je resursa potrebno uložiti da on prihvati njemu poslanu poruku. Kao što je prethodno objašnjeno, kada netko korisniku želi poslati poruku, on će prvo zatražiti njegov javni ključ. Korisnik će u poruci s javnim ključem kao dodatnu informaciju navesti i faktor težine kako bi pošiljatelj znao koliko zahtjevan mora biti dokaz za uspješan primitak poruke. Na taj način individualni korisnici imaju način za dodatno odvratiti pošiljatelje neželjene pošte.

3.7 *Namecoin* integracija

Kako su *Bitmessage* adrese prilično teške za pamćenje, *PyBitmessage* klijent nudi integraciju *Bitmessage* adresa s funkcionalnošću koju pruža kriptovaluta *Namecoin*. Tom integracijom moguće je povezati teško pamtljivu *Bitmessage* adresu s jednostavnim imenom koje pruža kriptovaluta *Namecoin*.

Namecoin je kriptovaluta nastala na temelju kriptovalute *Bitcoin* koja, osim za transakcije, koristi *blockchain* tehnologiju za spremanje dodatnih podataka. Ti dodatni podaci se primarno koriste na dva načina: kao temelj za alternativni sustav domenskih imena (DNS) te kao temelj za općenitu tehnologiju identifikacije putem lako pamtljivih imena.

Drugi način može se koristi i u ovom kontekstu – nakon integracije sa sustavom kriptovalute *Namecoin*, korisnik može u *Bitmessage* klijentu koristiti jednostavno ime, npr. *id/borna* ili samo *borna*, umjesto teško pamtljive *Bitmessage* adrese. Na slici 11 prikazana je konfiguracija *Namecoin* integracije u *PyBitmessage* klijentu.

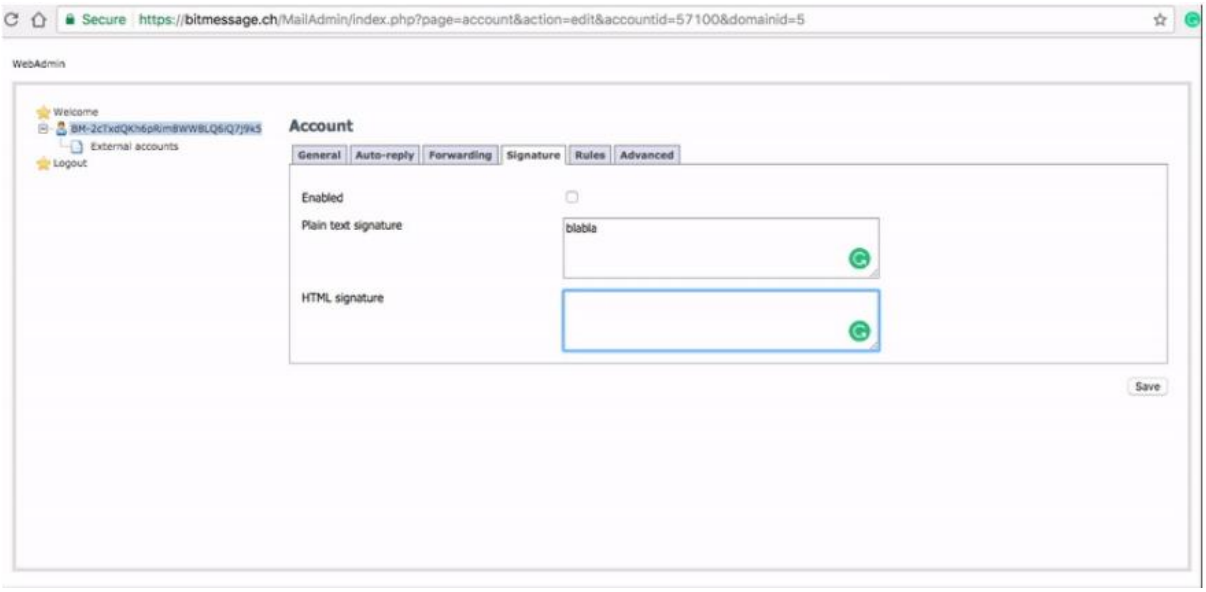


Slika 11 – konfiguracija *Namecoin* integracije u *PyBitmessage* klijentu

3.8 Programska podrška koja radi povrh *Bitmessagea*

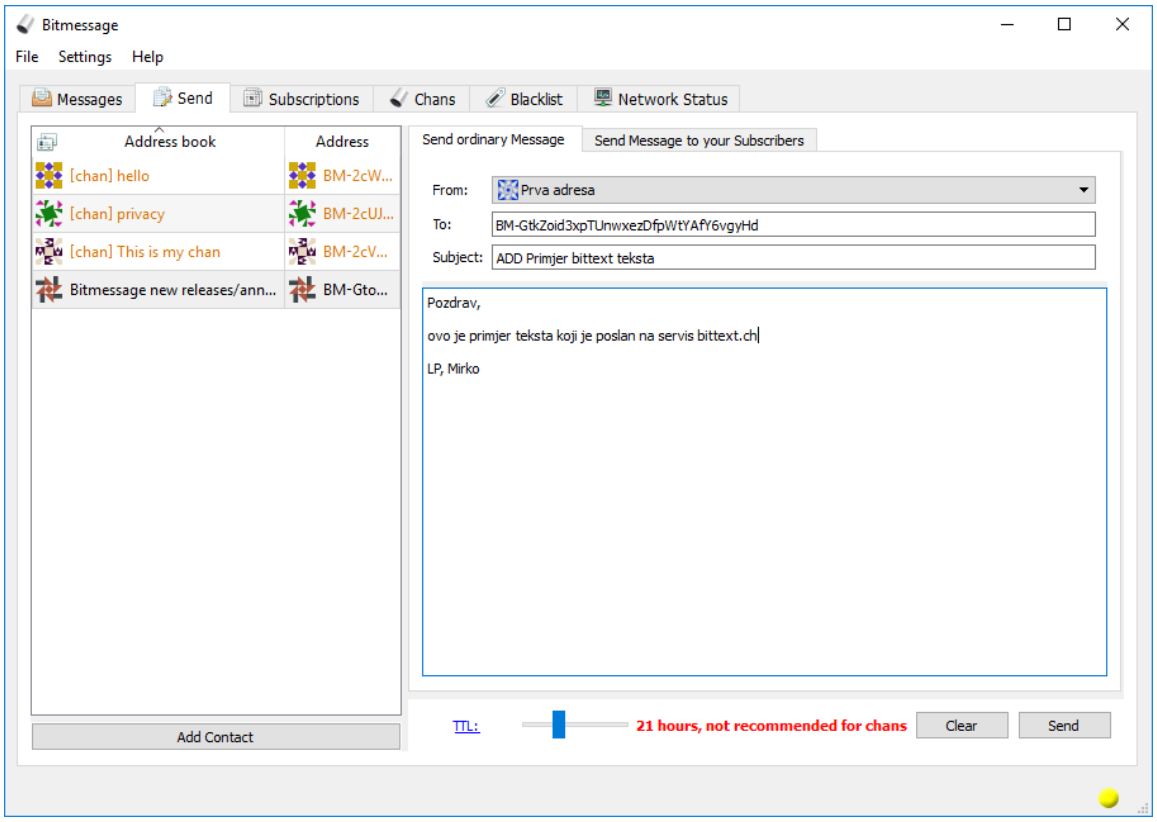
Korisnost *Bitmessagea* ne staje s prethodno navedenim značajkama – postoje razni servisi i alati koji pružaju dodatne funkcionalnosti čija se implementacija temelji na *Bitmessage* mreži.

Jedan takav popularan servis je Bitmessge.ch. *Bitmessage.ch* je servis koji povezuje *Bitmessage* mrežu i standardne sustave elektroničke pošte. Pomoću računa na *Bitmessage.ch* servisu moguće je slati te primati i *Bitmessage* poruke i poruke elektroničke pošte koristeći njihovo Web sučelje ili uobičajene klijente elektroničke pošte. Servis *Bitmessage.ch* praktičan je za korisnike koji žele slati poruke korisnicima *Bitmessage* mreže na jednostavan način, ali nešto manjom razinom sigurnosti i privatnosti. Na slici 12 prikazano je Web sučelje *Bitmessage.ch* servisa.

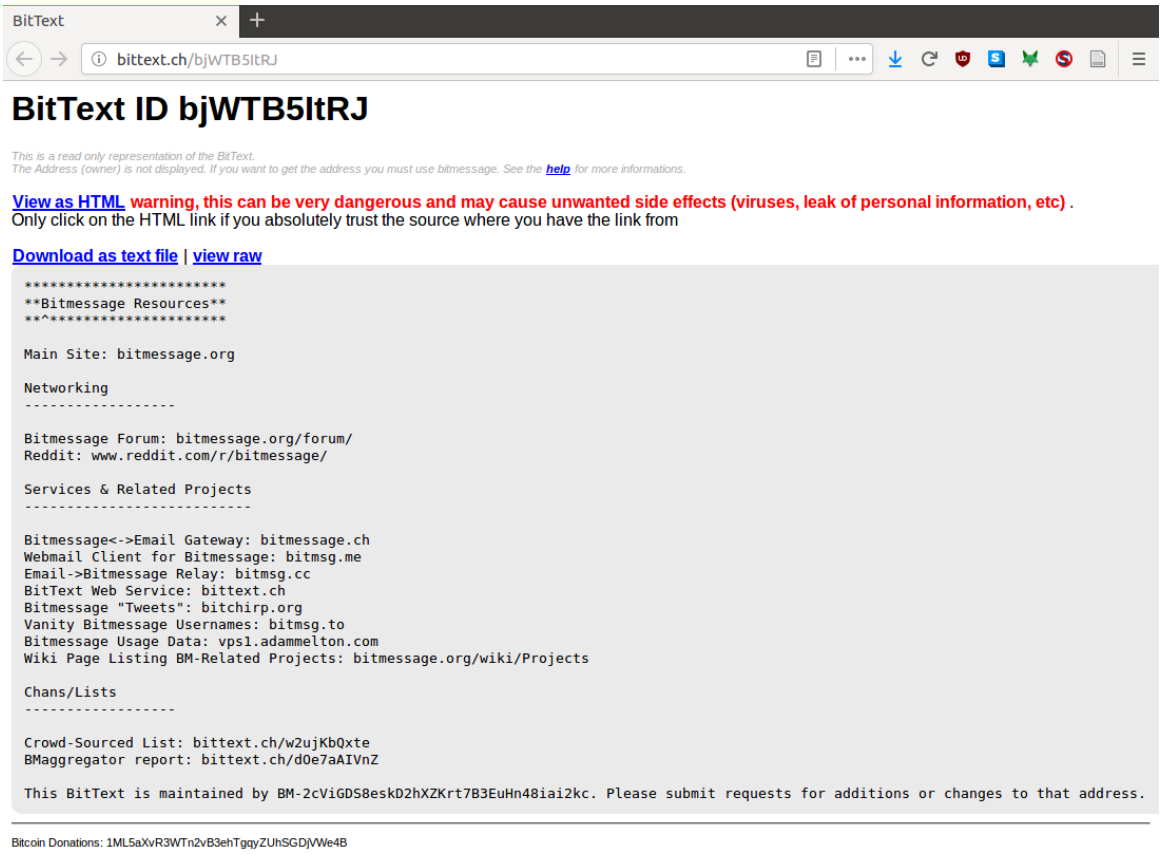


Slika 12 – Web sučelje *Bitmessage.ch* servisa (izvor)

Operateri *Bitmessage.ch* servisa održavaju i *BitText*, servis predstavljen kao *PasteBin* na *Bitmessage* mreži. *BitText* korisnicima omogućuje jednostavnu pohranu i pregledavanje tekstualnog sadržaja putem *Bitmessagea*. Servis se koristi slanjem naredbi na adresu *BM-GtkZoid3xpTUnwxezDfpWtYAfY6vgyHd* te je pohranjeni sadržaj dostupan za pregled i preko Web stranica servisa. Ključno je naglasiti kako je Web stranica zapravo samo sučelje za servis koji se temelji na *Bitmessageu* tako da bilo kakvi pokušaji napada ili cenzure Web stranica neće utjecati na dostupnost temeljnog servisa. Na slici 13 prikazan je postupak dodavanja teksta na *BitText* servis slanjem *Bitmessage* poruke s naredbom na odgovarajuću adresu, dok je na slici 14 prikazan pregled tekstualnog sadržaja putem Web sučelja.

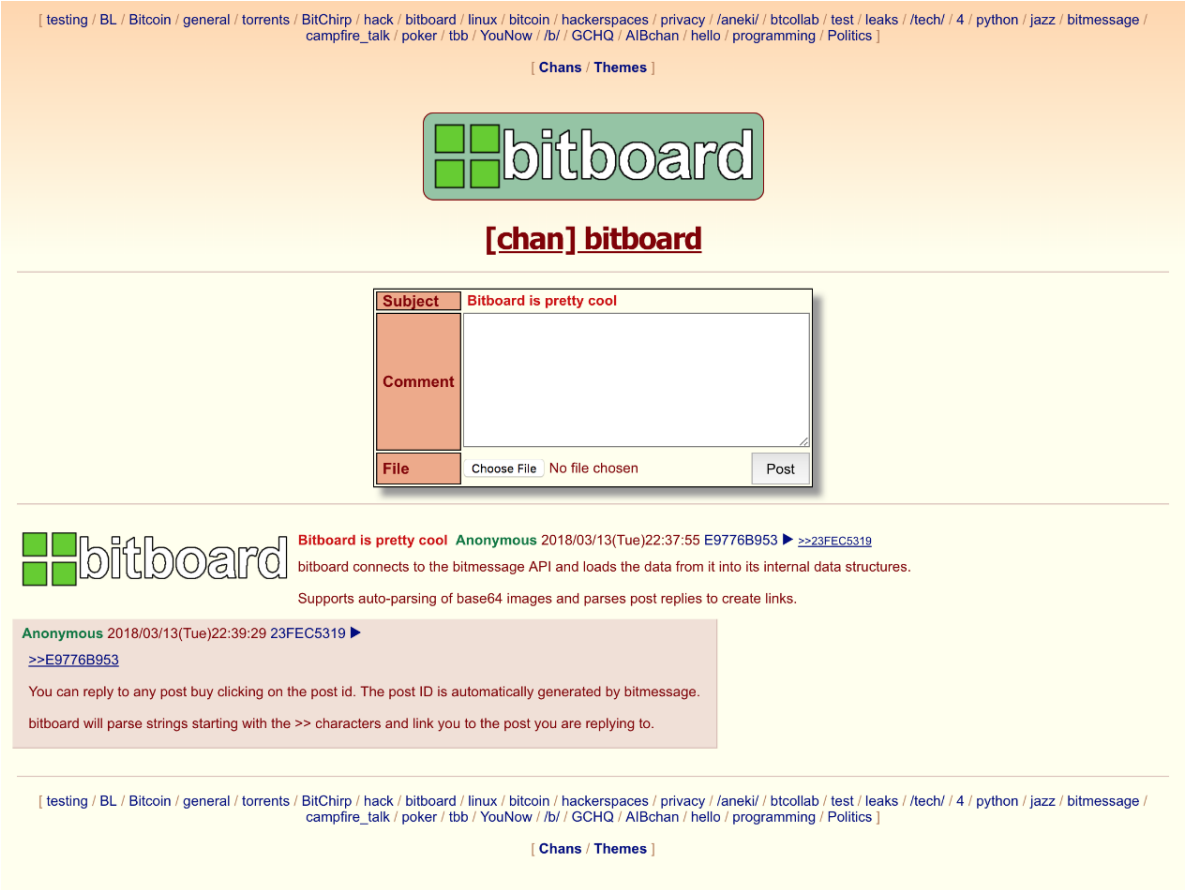


Slika 13 – dodavanje tekstualnog zapisa u *BitText* servis



Slika 14 – pregledavanje tekstualnog zapisa kroz Web sučelje *BitText* servisa

Još jedan servis temeljen na *Bitmessage* mreži je *Bitboard* – anonimna, decentralizirana platforma za razmjenu poruka i slika (eng. *imageboard*). *Bitboard* pruža bogato korisničko sučelje čiji se rad interno temelji na *Bitmessage* decentraliziranim listama razašiljanja. *Bitboard* sučelje prikazano je na slici 15.



Slika 15 – sučelje *Bitboard* platforme (izvor)

4 Zaključak

U nizu aplikacija i protokola za sigurnu i privatnu komunikaciju, *Bitmessage* se ističe kao jedinstveno rješenje. Načela inspirirana kriptovalutom *Bitcoin* omogućavaju *Bitmessageu* da na inovativni način riješi problem prikrivanja metapodataka te pojednostavni korisnicima razmjenu kriptografskih ključeva.

Trenutno, najveće mane *Bitmessagea* proizlaze iz toga što je on i dalje u fazi aktivnog razvoja. To ne znači da *Bitmessage* nije koristan u trenutnom obliku – u potpunosti ga je moguće koristiti za sve navedeno u ovom dokumentu, no on i dalje na više načina zaostaje za zrelijim rješenjima.

Primjerice, *Bitmessage* protokol i službeni klijent i dalje ne pružaju korisnicima jednostavan način slanja datoteka (eng. *attachment*). Trenutno, uobičajeni način slanja datoteka uključuje ručno kodiranje, slanje te dekodiranje datoteka *base64* kodom.

Još jedan nedostatak *Bitmessagea* koji će mnoge korisnike odvratiti od njegova korištenja je i njegova brzina. U trenutnom obliku, *Bitmessage* koristi osjetno veću količinu računalnih i mrežnih resursa u usporedbi s drugim rješenjima za sigurnu i privatnu komunikaciju. S pozitivne strane, razvojem protokola i klijenta potrebni resursi redovito se smanjuju te će se oni potencijalno i značajno smanjiti u budućnosti kada se korisnici krenu razdvajati u tokove.

Još jedno mjesto gdje *Bitmessage* ima mjesta za napredak je sigurnost službenog klijenta. Službeni klijent, *PyBitmessage*, i dalje nije prošao sigurnosni pregled koji su proveli stručnjaci, zbog nedostatka budžeta odnosno volontera koji bi mogli obaviti taj posao. Posljedice toga vidljive su u nedavnom napadu na *PyBitmessage* korisnike u kojem su napadači pronašli i iskorištavali ranjivost za udaljeno izvršavanje koda (eng. *remote code execution*) u inačici 0.6.2 *PyBitmessage* klijenta. Naravno, nije realno očekivati da aplikacija nema nikakve ranjivosti, no navedenu ranjivost korištenu u napadu bi zasigurno pronašli stručnjaci u sigurnosnom pregledu aplikacije.

S kriptografske strane, iako je *Bitmessage* impresivan jer rješava problem prikrivanja metapodataka, u *Bitmessageu* trenutno nije riješen problem osiguravanja buduće tajnosti (eng. *forward secrecy*). U kriptografiji pojam buduće tajnosti označava svojstvo protokola koje osigurava da kompromitacija dugoročnih ključeva ne ugrožava tajnost prošle komunikacije. Buduća tajnost je danas uobičajeno svojstvo protokola za sigurnu i privatnu komunikaciju te postoje prijedlozi i planovi za uvođenje buduće tajnosti u *Bitmessage*, no jednostavno zbog nedostatka radne snage to i dalje nije učinjeno.

Iako navedeni nedostaci nisu beznačajni, u dužem roku njih ne bi smjelo biti teško riješiti te u konačnici, *Bitmessage* je zaista jedinstveno rješenje za sigurnu i privatnu komunikaciju koje pokazuje izrazito velik potencijal.

5 Literatura

1. **Cole, David.** 'We Kill People Based on Metadata'. [Mrežno] 10. svibanj 2014. [Citirano: 20. lipanj 2018.] http://www.nybooks.com/daily/2014/05/10/we-kill-people-based-metadata.
2. **Warren, Jonathan.** Bitmessage: A Peer-to-Peer Message Authentication and Delivery System. [Mrežno] 2012. [Citirano: 20. lipanj 2018.] https://bitmessage.org/bitmessage.pdf.
3. **Šurda, Peter.** Improved stream handling · Bitmessage/PyBitmessage@f6bdad1. *GitHub.* [Mrežno] 6. veljača 2017. [Citirano: 28. lipanj 2018.] https://github.com/Bitmessage/PyBitmessage/commit/f6bdad18a36df91a681c5827e2e877f91bea6ce2.
4. **PyBitmessage.** PyBitmessage/api.py at c7917efbd9d6a92c66e5636348586f104f0871ca · Bitmessage/PyBitmessage. *GitHub.* [Mrežno] [Citirano: 28. lipanj 2018.] https://github.com/Bitmessaahttps://github.com/Bitmessage/PyBitmessage/blob/c7917efbd9d6a92c66e5636348586f104f0871ca/src/api.py#L407-L408ge/PyBitmessage.
5. **Thompson, Edward.** *In-depth analysis of Bitmessage.* [Mrežno] 2015. [Citirano: 24. lipanj 2018.] https://odinnsecurity.com/anonymity/in-depth-analysis-of-bitmessage/.
6. **reddit.** r/bitmessage - In case you didn't notice ; UPDATE PyBM right now! [Mrežno] 13. veljača 2018. [Citirano: 2. srpanj 2018.] https://www.reddit.com/r/bitmessage/comments/7xaplc/in_case_you_didnt_notice_update_pybm_right_now/.
7. **Johansen, Christian, i dr.** Comparing Implementations of Secure Messaging Protocols (long version). [Mrežno] studeni 2018. [Citirano: 28. lipanj 2018.] https://www.duo.uio.no/bitstream/handle/10852/60949/main_TR.pdf.
8. **Bitmessage Wiki.** Protocol specification. [Mrežno] [Citirano: 29. lipanj 2018.] https://www.bitmessage.org/wiki/Protocol_specification.
9. **Bitmessage Forum.** What's the difference between a chan, a mailing list, and a broadcast address? [Mrežno] [Citirano: 29. lipanj 2018.] https://bitmessage.org/forum/index.php?topic=3390.0.
10. **Kraft, Daniel.** NameID: Your Crypto-OpenID. [Mrežno] [Citirano: 29. lipanj 2018.] https://nameid.org/?view=faq.