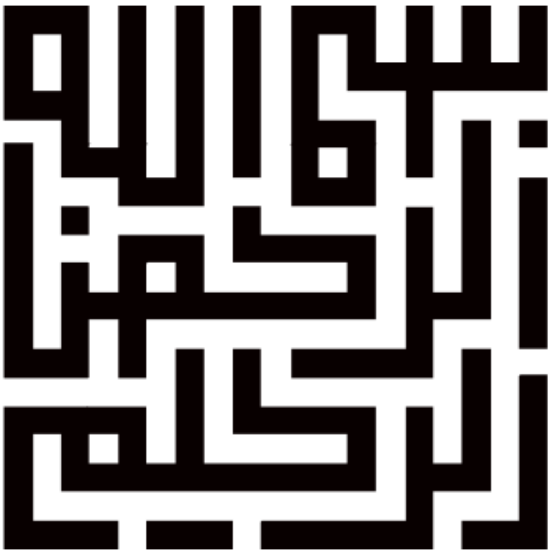
بنام خدا

بررسی معماری پیام‌رسان‌های اجتماعی مبتنی بر مدل ارتباطی نظیر به نظیر با تمرکز بر روی خصیصه‌های کیفی امنیت و حریم خصوصی



«وَإِذْ قَالَ إِبْرَاهِيمُ رَبِّ اجْعَلْ هَـٰذَا الْبَـلَدَ آمِنًا»

و هنگامی که ابراهیم گفت: «پروردگارا این شهر را شهری امن قرار ده»

(سوره ابراهیم، آیه ۳۵)

فهرست مطالب

فصل ۱ مقدمه

یکی امروزه رسانه‌های اجتماعی تبدیل به تکنولوژی فراگیر جهت ایجاد و تسهیل به اشتراک‌گذاری اطلاعات، ایده‌ها و علایق از طریق ارتباطات و تعاملات مجازی بر بستر شبکه شده است. طیف وسیعی از سرویس‌های ارائه شده در این حوزه ارائه یک تعریف مشخص از رسانه اجتماعی را با چالش مواجه کرده است با این وجود همه این سرویس‌ها دارای ویژگی‌های مشترکی نیز هستند. رسانه‌های اجتماعی، تعاملی و مبتنی بر وب هستند و محتوای آن توسط کاربران در قالب متن، نظر، عکس، صوت و ویدئو تولید می‌شود و تعاملات آنلاین کاربران نقش حیاتی و کلیدی در رسانه‌های اجتماعی دارد. کاربران پروفایل خود را ایجاد می‌کنند و رسانه اجتماعی با اتصال پروفایل کاربران به یکدیگر، توسعه شبکه‌های اجتماعی را تسهیل می‌کنند.

پیام‌رسان‌های اجتماعی یکی از سرویس‌های محبوب و فراگیر در حوزه رسانه‌های اجتماعی هستند که میلیون‌ها کاربر با کمک این نرم افزارها به ارتباط و تعامل با یکدیگر می‌پردازند. WhatsApp، Skype و تلگرام نمونه‌هایی از پیام‌رسان‌های اجتماعی هستند که ارتباط کاربران با یکدیگر را از طریق متن، صوت یا ویدئو را تسهیل می‌کنند. با این وجود دو دغدغه اصلی میان کاربران و توسعه دهندگان این سرویس‌ها وجود دارد که می‌تواند باعث برتری یک سرویس‌دهنده به سرویس‌دهنده دیگر شود. امنیت و کارایی دو نیاز کیفی مهم در سرویس‌های پیام‌رسان اجتماعی است. امنیت از دیدگاه افراد مختلف معانی مختلفی دارد و نمی‌توان یک روش ایده آل تحویل پیام برای همه سناریوها، همه کاربران و همه شرایط ارائه کرد. از طرفی مقیاس‌پذیر بودن سرویس به گونه‌ای که گسترش تعداد استفاده‌کنندگان از سرویس و توزیع آنها در گستره وسیع جغرافیایی نباید تاثیری بر کیفیت سرویس ارائه شده بگذارد.

از منظر مهندسی، برای هر کدام از این سناریوها، راه حل مشخصی وجود دارد تا با کمک آن بتوان مسائل و دغدغه‌های مختلف را بر طرف نمود. به عنوان مثال از منظر حریم خصوصی و دغدغه‌های امنیتی گاهی اوقات نیاز است تا هویت شما و فراداده‌های تولیدی به هنگام ارتباط با یک شخص دیگر، مخفی و پنهان بماند. همچنین هنگام هدایت ارتباطات، فراداده‌ها باید از دید سرورهایی که نقش رله را بازی می‌کنند پنهان بمانند. خیلی خوب می‌شود اگر کاربراز مسیری که پیام او از طریق آن هدایت می‌شود مطلع شده و در مورد آن مسیر و یا نوع اتصالات آن تصمیم‌گیری کند. کاربران باید بتوانند بدون هیچگونه اطلاعات قابل شناسایی و یا ردیابی و به صورت ناشناس ارتباط برقرار کنند.

این سند در چند بخش آماده شده است. ابتدا به بررسی شبکه‌های نظیر به نظیر می‌پردازیم و مدل‌ها و ویژگی‌های آن را شرح می‌دهیم. در ادامه سه نرم‌افزار پیام‌رسان مبتنی بر معماری نظیر به نظیر را معرفی می‌کنیم. ابتدا به بررسی نرم‌افزار پیام‌رسان Bleep می‌پردازیم که توسط شرکت BitTorrent ارائه شده است و معماری و نحوه کارکرد آن را معرفی می‌کنیم. سپس نرم‌افزار پیام‌رسان Tox را شرح داده و ماژول‌های تشکیل دهنده آن را که هسته Toxرا تشکیل می‌دهند را معرفی می‌کنیم. در پایان، نرم‌افزار پیام‌رسانی Briar را معرفی کرده و پروتکل‌های آن را توضیح می‌دهیم.

.

فصل ۲ شبکه‌های نظیر به نظیر

به شبکه‌های نظیر به نظیر حول این مفهوم ایجاد شده است که بر خلاف ساختارهای مشتری-خدمتگزار یک گره به طور همزمان هم نقش مشتری را بازی می‌کند و هم نقش سرور را برای دیگر گره‌ها ایفا می‌کند. این شبکه‌ها معمولا یک شبکه مجازی همپوشان بر روی شبکه فیزیکی زیرین ایجاد می‌کنند به گونه‌ای که گره‌های شرکت‌کننده در ایجاد این شبکه همپوشان، زیرمجموعه‌ای از گره‌ها در شبکه فیزیکی هستند. اگرچه داده‌ها از طریق زیرساخت فیزیکی و با کمک پروتکل‌های TCP و UDP منتقل می‌شوند اما در لایه Application این امکان وجود دارد که گره‌ها به طور مستقیم از طریق لینک‌های مجازی با یکدیگر در ارتباط باشند. ایجاد این شبکه همپوشان مجازی به این دلیل است که شاخص‌گذاری و اکتشاف گره‌ها بتوانند مستقل از توپولوژی شبکه فیزیکی صورت پذیرد. بر اساس اینکه گره‌ها به چه صورت به یکدیگر متصل هستند و منابع چگونه شاخص‌گذاری و پیدا می‌شوند، شبکه‌های مجازی همپوشان را می‌توان به دو دسته شبکه‌های ساختارمند و فاقدساختار تقسیم کرد.

۲-۱ شبکه‌های فاقد ساختار

این دسته از شبکه‌ها هیچگونه ساختاری را در طراحی شبکه همپوشان مجازی در نظر نمی‌گیرند و گره‌های موجود در این شبکه به شکل تصادفی با یکدیگر اتصال برقرار می‌کنند. Gnutella، Gossip و Kazaa مثال‌های از پروتکل نظیر به نظیر فاقد ساختار است. ساخت این نوع شبکه‌ها آسان است و امکان بهینه‌سازی در نواحی مختلف این شبکه همپوشان را فراهم می‌کند. همچنین در برابر پیوستن و جدا شدن مکرر گره‌ها[1] از استحکام بالایی برخوردار است. اما اصلی‌ترین محدودیت این نوع شبکه‌ها، فقدان ساختار در آنهاست. به عنوان نمونه، وقتی که می‌خواهیم داده‌ای را در شبکه پیدا کنیم پرس وجو باید شکل سیل‌آسا در شبکه پخش شود تا بیشترین گره‌هایی که این داده را به اشتراک می‌گذارند پیدا شوند. این مساله باعث ترافیک شبکه و مصرف منابع پردازشی و حافظه می‌گردد واین اطمینان وجود ندارد که جستجو به سرانجام برسد. همچنین از آنجایی که هیچ ارتباطی بین گره‌ها و داده‌هایی که مدیریت می‌کنند وجود ندارد، هیچ تضمینی وجود ندارد که داده مورد نظر ما در اختیار یک گره قرار داشته باشد.

۲-۲ شبکه‌های ساختارمند

در این دسته، شبکه همپوشان در قالب یک توپولوژی مشخص، سازماندهی می‌شود و این اطمینان وجود دارد که گره‌ها به شکل بهینه‌ای، فایل یا منابع مورد نیاز خود را در شبکه جستجو کنند حتی اگر آن منبع بسیار کمیاب باشد. اکثر شبکه‌های نظیر به نظیر ساختارمند، با کمک DHT[2]، ماالکیت منابع را به گره‌های شبکه تخصیص می‌دهند. در DHT، منابع در قالب زوج کلید - مقدار ذخیره می‌شوند که این امکان را به گره‌ها می‌دهد تا منبع مورد نظر را با کمک یک جدول هش مورد جستجو قرار داده و مقدار متناظر به کلید مربوطه را بازیابی کنند. با این وجود، برای هدایت موثر ترافیک در شبکه، گره‌ها در یک شبکه همپوشان ساختارمند، باید لیستی از همسایگان خود که شاخص‌های مشخصی را محقق می‌کنند را داشته باشند. همین مساله باعث می‌شود که این شبکه‌ها در برابر نرخ بالای پیوستن و جداشدن گره‌ها از شبکه، از استحکام کمتری برخوردار باشند. Chord نمونه‌ای مشهور از پروتکل نظیر به نظیر است که از DHT استفاده کرده است. همچنین

---

[1] Churn

[2] Distributed Hash Table (DHT)

BitTorrent یکی از مشهورترین سرویس‌های به اشتراک‌گذاری فایل مبتنی بر شبکه‌های نظیر به نظیر است که از DHT استفاده کرده است.

۲-۳ مدل‌های ترکیبی

این مدل، تلفیقی از مدل‌های نظیر به نظیر و مشتری – خدمت‌گذار را ارائه می‌کند. به طور معمول این مدل‌ها دارای یک سرور مرکزی است که به گره‌ها کمک می‌کند تا یکدیگر را پیدا کنند. Spotify نمونه‌ای از این مدل‌هاست. انواع مختلفی از این مدل‌های ترکیبی وجود دارد که یک مصالحه بین ارائه متمرکز کارکردها از طریق شبکه ساختارمند مشتری-خدمت‌گذار از یک سو و برابری گره‌ها در شبکه نظیر به نظیر محض و فاقد ساختار، برقرار می‌کند. مدل‌های ترکیبی کارایی بهتری نسبت به هر یک از مدل‌ها به تنهایی ارائه می‌کند زیرا در برخی از کارها نظیر جستجو نیاز به ارائه کارکردهای متمرکز است در حالی که می‌توان از مزایای تجمع غیر متمرکز گره‌ها در شبکه‌های فاقد ساختار نیز بهره‌مند شد.

۲-۴ امنیت و اعتماد

شبکه‌های نظیر به نظیر در معرض چالش‌های زیادی از منظر امنیت هستند. مشابه سایر گونه‌های نرم افزاری، کاربردهای مبتنی بر P2P در معرض آسیب‌پذیری هستند اما آنچه که این مساله را جدی و خطرناکتر می‌کند این است که آنها هم‌زمان با ایفای نقش مشتری، می‌توانند در قالب سرور نیز ظاهر شوند. این مساله آنها را در برابر کدهای مخرب راه دور[3] آسیب‌پذیرتر می‌کند.

از آنجایی که هر گره در هدایت ترافیک شبکه نقش ایفا می‌کند، کاربران بدخواه می‌توانند از این امکان برای اجرای طیف مختلفی از حملات مسیریابی[4] و یا حملات محروم سازی از سرویس[5] استفاده‌کنند. به عنوان مثال یکی از این حملات، مسیر یابی غلط[6] است که در آن گره‌های تخریب‌گر، عامدانه درخواست‌ها را به اشتباه هدایت می‌کنند و یا false برمی‌گردانند. حمله دیگر، به روزرسانی اشتباه اطلاعات مسیریابی[7] است که طی آن، گره‌های تخریب‌گر با ارسال داده‌های اشتباه به گره‌های همسایه، باعث آلوده شدن اطلاعات جدول مسیریابی آنها می‌شوند. یکی دیگر از این حملات زمانی اتفاق می‌افتد که گره‌ای از طریق یک گره مخرب، به شبکه می‌پیوندد [8]که باعث می‌شود این گره جدید، در بخشی[9] از شبکه قرار گیرد که در آن سایر گره‌های مخرب حضور دارند.

نرخ شیوع بدافزارها در شبکه‌های مختلف P2P متفاوت است. به عنوان مثال ۶۳ درصد از درخواست‌های دانلود پاسخ داده شده، حاوی گونه‌هایی از بدافزارهستند در حالیکه ۳ درصد از محتوای OpenFT حاوی بدافزار می‌باشد. همچنین داده‌های آلوده نیز از طریق تغییر فایل‌هایی که به اشتراک گذاشته شده‌اند در شبکه P2P می‌تواند توزیع شود.

این مسائل باعث شد تا شبکه‌های نظیر به نظیر با استفاده از روش‌های جدید هش کردن، اعتبارسنجی و رمزنگاری امنیت فایل‌ها و مکانیسم‌های اعتبارسنجی خود را به شکل بسیار زیادی بهبود بخشند به گونه‌ای که تقریبا در برابر اکثر حملات

---

[3] Remote exploit
[4] Routing attcacks
[5] Denial of service attacks
[6] Incorrect Lookup routing
[7] Incorrect routing update
[8] Bootstrap
[9] Partition

مقاومت کنند حتی اگر بخش اصلی شبکه توسط میزبان‌های جعلی و یا غیر عملیاتی جایگزین شوند. طبیعت غیر متمرکز شبکه‌های نظیر به نظیر استحکام آنها در ارائه سرویس را بهبود بخشیده است زیرا مشکل وجود یک نقطه شکست[10] در سیستم‌های مبتنی بر مدل مشتری – خدمت‌گذار را ندارند. هم‌زمان با پیوستن یک گره به شبکه و افزایش تقاضاها، ظرفیت شبکه نظیر به نظیر نیز افزایش می‌یابد و احتمال شکست کاهش می‌یابد. اگر یک گره در ارائه کارکردهای خود دچار شکست شود بقیه شبکه در معرض خطر نخواهد بود و آسیب نخواهد دید.

۵-۲ ذخیره‌سازی و جست‌وجو

شبکه‌های نظیر به نظیر از منظر پایداری و دسترس‌پذیری در مقایسه با شبکه‌های متمرکز دارای مزایا و معایبی هستند. در شبکه‌های متمرکز، مدیر سیستم کنترل و مدیریت را در دست دارد و بنابراین او می‌تواند نسبت به میزبانی یا حذف یک محتوا تصمیم‌گیری کند و در نتیجه در یک سیستم پایدار خیال کاربران از منظر دستیابی به محتوایی که میزبانی شده است راحت است.

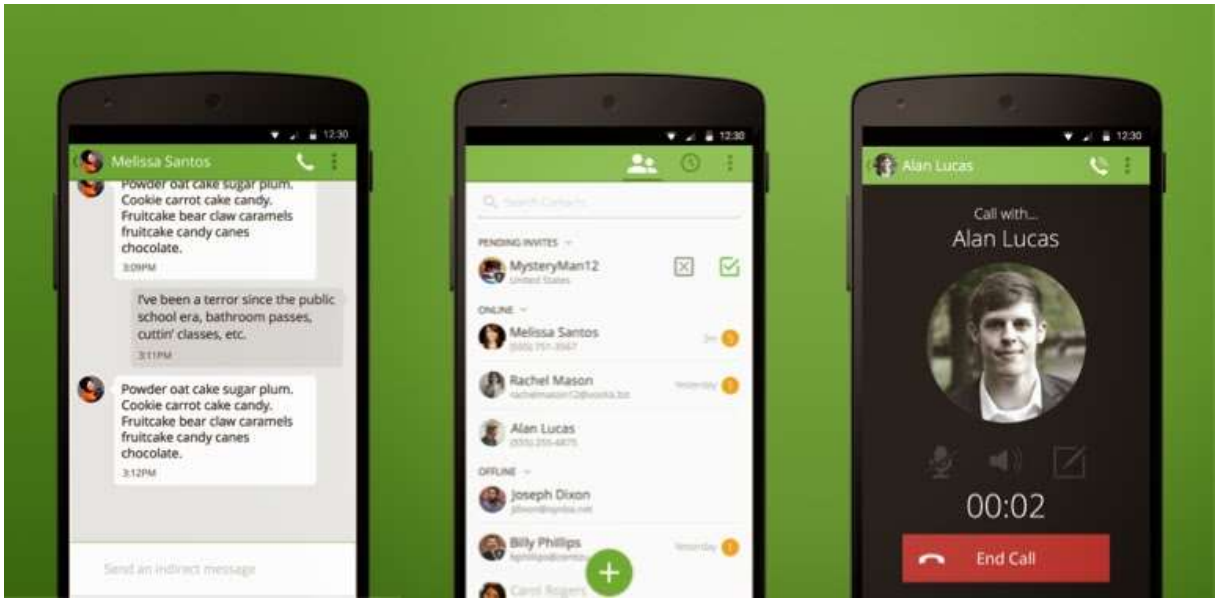
در شبکه‌های نظیر به نظیر، این جامعه کاربران هستند که تصمیم می‌گیرند چه محتوایی را به اشتراک بگذارند. بنابراین محتوای غیر مشهور، به تدریج محو می‌شود زمانی که کاربران آن را به اشتراک نگذارند، دسترس‌پذیری آن نیز متوقف می‌شود. در حالیکه محتوا و فایل‌های مشهور به شکل بسیار گسترده‌ای توزیع و به اشتراک گذاشته می‌شوند به گونه‌ای که دسترس‌پذیری به آن بسیار بیشتر از حالت متمرکز است. در حالت متمرکز کافی است لینک ارتباطی با سرور قطع شود در این صورت دسترسی به محتوای میزبانی شده مختل می‌گردد. در حالیکه در شبکه‌های نظیر به نظیر از دست رفتن ارتباط با یک گره مانع از دستیابی به یک فایل مشهور نمی‌شود زیرا این محتوا بر روی گره‌های مختلفی توزیع و به اشتراک گذاشته شده است.

فقدان یک کنترل و مدیریت مرکزی در شبکه‌های نظیر به نظیر، آنها را در برابر فشار نهادها و دولت‌ها نیز مقاوم تر می‌کند و آنها نمی‌توانند محتوایی را حذف و یا مانع از به اشتراک گذاری آن شوند. در حالیکه در شبکه‌های متمرکز مدیریت سیستم می‌تواند تحت فشار برای حذف یک محتوا قرار گیرد. از منظر بازیابی و پشتیبان‌گیری، در سیستم‌های متمرکز این وظیفه بر عهده مدیر سامانه قرار دارد در حالیکه در شبکه توزیع شده هر گره مسئول پشتیبان‌گیری و بازیابی خود است.

[10] Single point of failure

فصل ۳ نرم‌افزار پیام‌رسان Bleep

امنیت داده‌ها یکی از دغدغه‌های اصلی کاربران هنگام استفاده ازسرویس‌هایی است که بر بستر وب یا فضای ابری ارائه می‌شود. بر همین اساس شرکت BitTorrent به عنوان مخترع پروتکل BitTorrent که امروزه از محبوب‌ترین پروتکل‌های توزیع و اشتراک‌گذاری نظیر به نظیر[11] است، سرویس پیام‌رسان کاملا رمزنگاری شده‌ای را با نام Bleep در سال ۲۰۱۵ ارائه کرد که مبتنی بر همین پروتکل است و در آن پیام‌ها و داده‌ها به صورت محلی و در دستگاه کاربران ذخیره می‌شود، با این ویژگی که امکان حذف پیام و تاریخچه آن بدون برجای گذاشتن هیچ ردی وجود دارد. قابل توجه است که با توجه به مشکلات مالی شرکت BitTorrent کار فعال بر روی این پیام‌رسان متوقف شده است اما درس‌آموخته‌ها و الگوهای معماری به کار رفته در آن می‌توان نقشه راهی برای محققین در طراحی و ساخت راه‌کارهای مشابه باشد.



این تصور مطرح است که نرم افزارهای پیام رسان مبتنی بر معماری توزیع شده نظیر به نظیر، امنیت بیشتری نسبت به نرم افزارهای مبتنی بر معماری ابری دارند. Jaehee Lee یکی از مدیران تولید BitTorrent می‌گوید «در سرویسهای مبتنی بر ابر، اطلاعات شخصی و خصوصی کاربران در سرورها ذخیره می‌شوند که به طور بالقوه در برابر حملات، آسیب پذیر هستند». معماری Bleep، فاقد سرور است و با کمک DHT[12] محقق شد و فارغ از اینکه محتوای پیام چگونه ارسال می‌شود، نرم افزار از شبکه توزیع شده کاربران برای پیدا کردن یکدیگر استفاده می‌کند. برای استفاده از این سرویس کاربران می‌توانند به صورت ناشناس و یا با کمک شماره تلفن یا ایمیل وارد شوند و سایر دوستان خود را از طریق ایمیل، SMS، QR code و یا یک کلید عمومی[13] دعوت کنند.

---

[11] Peer-to-Peer
[12] Distributed hash table
[13] Public key

۳-۱ نحوه کار Bleep

در این بخش تلاش می‌کنیم تا با برخی از کارکردهای اصلی Bleep آشنا شویم. به همین منظور نحوه ایجاد هویت برای کاربران وشیوه احراز آن، شیوه پیوستن گره‌ها به شبکه DHT، چگونگی دعوت از افراد و راه‌اندازی یک کانال امن ارتباطی را به اختصار شرح می‌دهیم.

۳-۱-۱ ایجاد هویت و نحوه احراز آن

وقتی کاربر برای اولین بار نرم افزار را نصب می‌کند یک کلید خصوصی برای آن تولید می‌شود که قابل استفاده در دستگاه‌های مختلف است و تحت حساب کاربری[14] کاربر، رمز می‌شود تا سایر کاربران محلی این دستگاه نتوانند به آن دسترسی داشته باشند. همه کاربران در ابتدا به صورت ناشناس ثبت نام می‌شوند اما اگر کاربر خواهان تائید هویت خود از طریق ایمیل یا شماره تلفن باشد آنگاه یک توکن از طریق سرور احراز هویت برای او ارسال می‌شود تا مطمئن شویم او همان فردی است که ادعای آن را می‌کند. کلید عمومی از روی کلید خصوصی تولید شده و در سرور ثبت می‌شود. این کلید عمومی فقط برای زمانی است که می‌خواهید دوستان خود را از طریق ایمیل یا شماره تلفن پیدا کنید و یا کسی را به عنوان دوست خود اضافه کنید. این فرایند جستجو فقط هنگامی که کاربر، فردی را به لیست تماس خود اضافه می‌کند انجام می‌شود و بعد از آن، همه جستجوها از قبیل یافتن آدرس IP فرد توسط DHT انجام می‌شود. با کمک یک پروتکل قابلیت آشکارکردن گراف دوستی در BitTorrent محدود شده است. کلید عمومی کاربرانی که به عنوان ناشناس ثبت نام کرده اند، در سرور رجسیتر نمی‌شود و آنها از کد QR و یا کلیدهای عمومی که مستقیما اضافه کرده اند استفاده می‌کنند.

۳-۱-۲ پیوستن به شبکه DHT

Bleep مبتنی بر DHT است اما توزیع شدگی به تنهایی تضمین کننده امنیت و محرمانگی اطلاعات کاربران نیست به همین دلیل یکی از ویژگی‌های مهم برای تحقق امنیت این است که ترافیک DHT، از کلید عمومی کاربر جدا شده است. بنابراین اگر دوست یک کاربر نباشید، فهمیدن این مساله که یک کلید عمومی، متناظر با چه آدرس IP می‌باشد امر بسیار دشواری است. همچنین به لحاظ عملیاتی غیر ممکن است که بفهمیم چه کسی با چه کسی و در چه زمانی در حال محاوره است حتی اگر حمله‌کننده به اطلاعاتی از جنس اطلاعات ISPها دسترسی داشته باشد.

موتور Bleep تعدادی از گره‌ها را به صورت تصادفی برای مشارکت در DHT انتخاب می‌کند. نکته مهم اینجاست که تعدادگره‌هایی که در DHT مشارکت دارند بر روی حریم خصوصی کاربر تاثیر دارد. اگر تعداد گره‌های مشارکت کننده در DHT کم باشد، شما می‌توانید گره‌های زیادی را به شبکه اضافه کنید و بخشی از اطلاعات را جمع آوری کنید که منجر به نشت[15] فراداده‌ها شود.

۳-۱-۳ محافظت از فراداده‌ها

فراداده‌ها مشخص می‌کند که چه کسی با چه کسی و در چه زمانی در حال محاوره است. در Bleep یک مخزن مرکزی برای نگهداری این اطلاعات وجود ندارد و معماری آن، کار را برای دسترسی به ترافیک شبکه وجمع آوری فراداده‌های مورد نظر دشوار می‌کند.

---

[14] User account
[15] Leakage

۳-۱-۴ دعوت افراد به صورت خصوصی

وقتی کاربر، فردی را به لیست تماس خود اضافه می‌کند، دعوت نامه برای آن فرد ارسال می‌شود تا مشخص شود که آیا مایل است، پیام دریافت کند و وضعیت آنلاین بودن او مشخص شود یا نه. این دعوت نامه رمزگذاری می‌شود تا فقط کاربری که مد نظر است آن را بتواند بخواند. وقتی هر دو کاربر دارای کلید عمومی یکدیگر باشند، آنگاه اتصال مستقیم بین آنها از طریق یافتن آدرس IP و تخصیص پورت روی شبکه DHT میسر خواهد بود.

۳-۱-۵ راه‌اندازی یک کانال امن

وقتی کاربر دعوت نامه یک دوست را پذیرفت یک کانال رمزگذاری شده مبتنی بر UDP بین دو گره نظیر به نظیر شکل می‌گیرد. پیامی که در این کانال ارسال می‌شود به صورت end-to-end رمزگذاری شده است. همچنین کلید رمزگذاری در طول زمان تغییر می‌کند تا امکان رمز گشایی ترافیک بیش از گذشته دشوار شود.

۳-۱-۶ بررسی معماری Bleep از منظر امنیت

یکی از نقاط ضعف رویکردهای موجود در نرم افزارهای پیام رسان، اتکای آن به سرورهای مرکزی، جهت هدایت و ذخیره سازی پیامهای ارتباطی است. بنابراین حتی اگر آنها استانداردهای صنعتی در حوزه امینت را رعایت کرده باشند نمی‌توانند تضمین کنند که ارتباطات شما امن و محفوظ خواهد ماند. فقط کافی است به سرورهای مرکزی تامین کننده سرویس ارتباطی نفوذ کرد تا مفهوم حریم خصوصی در معرض خطر جدی قرار گیرد.

اما سوالی که پیش می‌آید این است که چگونه ارتباطات بدون حضور سرویس دهنده‌های مرکزی انجام می‌شود و فرایند ورود[16] جهت بهره‌گیری از این سرویس چگونه است و نام کاربری و رمز عبور کجا ذخیره سازی می‌شود؟ اتصال بین کاربران چگونه برقرار می‌شود و ارتباطات به چه شکل هدایت می‌شوند؟ معماری Bleep دارای ۲ مولفه اصلی است:

۱. پلتفرم ارتباطی نظیر به نظیر که می‌توان آن را در قالب سرور SIP[17] کاملا توزیع شده تصور کرد که در حکم موتور Bleep عمل می‌کند.
۲. واسط کاربری برای ارتباط متنی و صوتی که یک UAC[18] سازگار با SIP است.

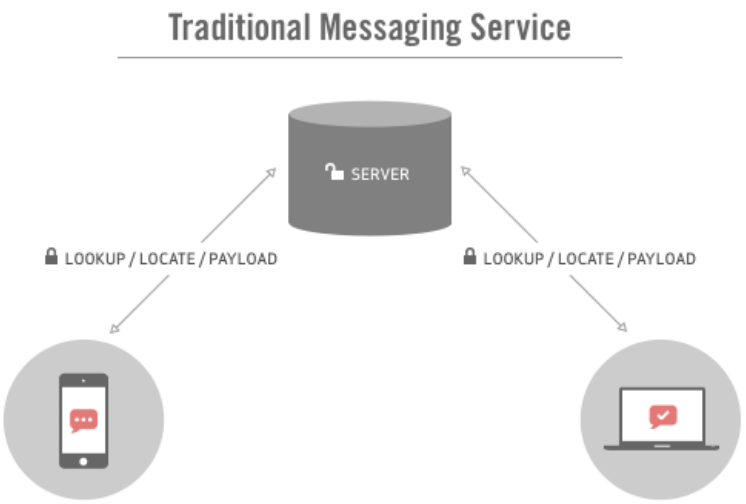
اگر چه بسیاری از نرم افزارها برای بهبود امنیت از روش رمزنگاری end-to-end استفاده می‌کنند اما نحوه اداره کردن فراداده‌ها در آنها می‌تواند باعث افشای برخی از اطلاعات محرمانه شود. در مقابل برخی از ویژگی‌هایی که باعث بهبود امینت و محرمانگی اطلاعات در Bleep می‌شود عبارتند از:

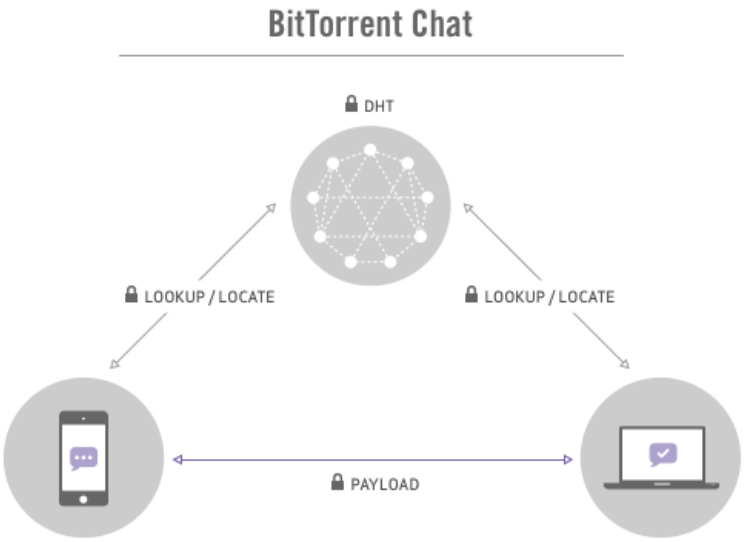
۱. هیچ مخزن مرکزی برای نگهداری فراداده‌ها وجود ندارد و BitTorrent اطلاعاتی از قبیل اینکه چه کسی با چه کسی ارتباط برقرار کرده است را ردیابی و یا ذخیره نمی‌کند حتی به صورت موقتی.
۲. هیچ سرور مرکزی برای انجام جستجوها وجود ندارد و کاربران با کمک سایر گره‌ها، فردی را که به دنبال آن هستند را پیدا می‌کنند و اطلاعاتی در مورد اینکه یک کاربر به دنبال چه کسی می‌گردد، ذخیره نمی‌شود.

---

[16] Login
[17] Session Initiation Protocol
[18] User Agent Client

۳. همه لینکها با کمک پروتکل‌هـایی نظیـر curve25519, ed25519 , salsa20, poly1305 رمزگـذاری شـده است.





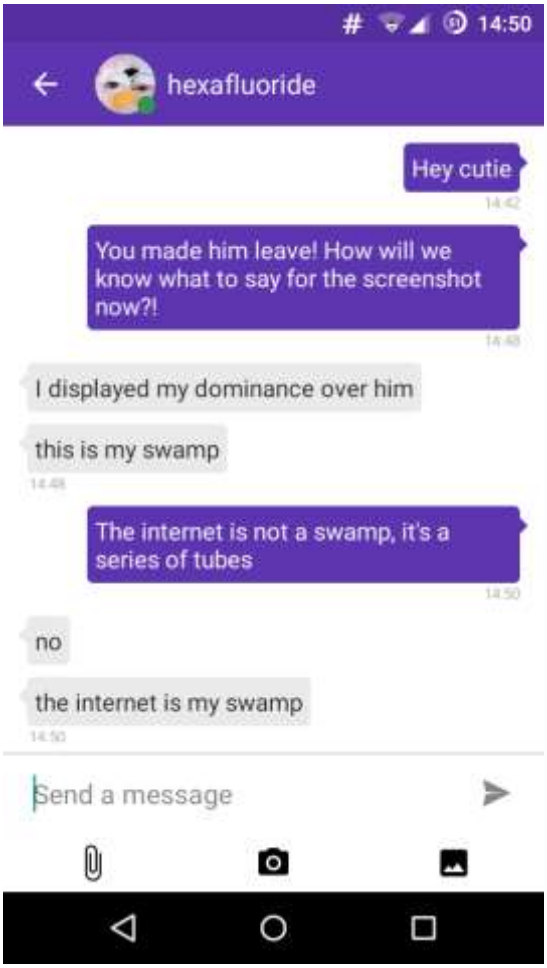
در نرم افزار پیام‌رسانی BitTorrent هیچگونه نام کاربری وجود ندارد و فرایند ورد به صورتی که در سایر سرویس‌دهنده‌ها با آن آشنا هستیم نیست. هر فردی که روی شبکه پیام رسانی BitTorrent قرار دارد، دارای یک کلید عمومی[19] است. در نتیجه شما می‌توانید بدون اینکه هویت خود را آشکار کنید به یک نفر پیام ارسال کنید. دو نفر برای ارتباط فقط نیازمند اطلاع از کلید عمومی یکدیگر هستند. استفاده از رمزگذاری با استفاده از کلید عمومی مزایای زیادی دارد. مهمترین آن این است که می‌توانید پیام خود را با استفاده از کلید خصوصی خود و کلید عمومی گیرنده پیام رمز کنید. پیام‌های ارسالی برای شما فقط از طریق کلید خصوصی شما قابل رمز گشایی است. برای امنیت بیشتر، یک کلید رمز گذاری موقتی بر اساس زوج کلیدهای شما تولید می‌شود که فقط مختص به این محاوره است و پس از این محاوره پاک می‌شود.

زیرساخت این پیام‌رسان مبتنی بر روشی است که کلید عمومی به آدرس IP ترجمه می‌کند و برای این کار از DHT استفاده می‌کند. این مساله باعث می‌شود که برای هدایت پیامها نیازمند سرورهای مرکزی نباشیم. DHT شبکه‌ای ازگره‌های همکار است که با شما در یافتن آدرس فردی را که دنبال ارسال پیام به او هستید مشارکت می‌کند. برای این منظور شما پرس و جو را از نزدیک ترین همسایگان خود شروع می‌کنید و در صورت پیدا نشدن آدرس فرد مورد نظر، این کار را در مورد همسایه همسایگان خود ادامه می‌دهید تا بالاخره به گرهی برسید که آدرس فرد مورد نظر شما را داشته باشد و این آدرس به شما برگشت داده می‌شود.

19

فصل ۴ نرم‌افزار پیام‌رسان Tox

نرم‌افزار پیام‌رسان Tox بعد از افشاگری‌های ادوارد اسنودن در مورد فعالیت‌های جاسوسی NSA کار خود را شروع کرد. ایده کار بر این اساس بود که نرم‌افزارپیام‌رسانی ایجاد شود که متکی و نیازمند به یک سرور مرکزی نباشد. سیستم باید توزیع‌شده و مبتنی بر معماری نظیر به نظیر می‌بود و از رمزنگاری سراسری استفاده می‌کرد به گونه‌ای که نتوان هیچ‌کدام از ویژگی‌های رمزنگاری را در آن غیرفعال کرد. با این وجود سیستم باید به آسانی قابل استفاده می‌بود، بدون اینکه کاربر نیازمند دانش خاصی در زمینه رمزنگاری و سیستم‌های توزیع شده داشته باشد.



درسال ۲۰۱۳ یک گروه کوچک از توسعه‌دهندگان از سراسر دنیا تشکیل شد و کار روی پیاده‌سازی پروتکل Tox را در قالب یک کتابخانه آغاز کرد. این کتابخانه همه تسهیلات لازم برای پیام‌رسانی و رمزنگاری را فراهم می‌کرد و مستقل از هر واسط کاربری بود. کاربر نهایی برای بهره‌گیری از Tox نیازمند Tox Client است. Tox یک پروژه متن‌باز و رایگان است که توسط توسعه دهندگان داوطلب به پیش می‌رود و وابسته به یک شرکت یا سازمان نیست. در طی این مدت، پروژه‌های مستقل و متعددی از Tox Client ایجاد شده است و کتابخانه‌ای که هسته Tox را پیاده‌سازی می‌کرد به طور پیوسته در حال بهبود و ارتقاء است.

iOS	Android	OSX	BSD	GNU/Linux	Windows	

qTox	✓	✓	✓	✓		
uTox	✓	✓	✓	در حد کمینه		
Ricin	به زودی	✓	✓			
Toxygen	✓	✓				
Toxic		✓	✓	✓		
Antox					✓	
Antidote						✓

۴-۱ اهداف Tox

Tox تلاش می‌کند تا ارتباط برقرار شده بین کاربران را تأیید کند. این بدان معنی است که در طول یک جلسه ارتباطی، هر دو طرف می‌توانند از هویت طرف دیگر مطمئن شوند. کاربران با کلید عمومی خود شناسایی می‌شوند. اگر کلید مخفی[20] به خطر بیافتد، هویت کاربر به خطر می‌افتد و مهاجم می‌تواند هویت این کاربر را جعل کند. در چنین شرایطی راه حل این مشکل، ایجاد یک هویت جدید با استفاده از یک کلید عمومی جدید است.

یکی دیگر از اهداف Tox استفاده از رمزگذاری سراسری[21] است. پروتکل Tox، لینک‌های ارتباطی با قابلیت رمزگذاری سراسری را ارائه می‌کند که در آن کلید اشتراکی با استفاده از روشی مشابه Diff-Hellman به شکل قطعی به دست می‌آید و در نتیجه نیاز به ارسال کلیدها بر روی شبکه وجود ندارد. همچنین بعد از برقراری اتصال بین گره‌های نظیر به نظیر، بر روی کلیدهای نشست، مذاکره مجدد انجام می‌شود. Tox هنگام درخواست دوستی با کمک یک پروتکل مسیریابی پیازی[22] می‌تواند از یافتن آدرس IP یک گره موردی با استفاده از کلید عمومی توسط نفوذگران جلوگیری کند

Tox مستقل از زیر ساخت است و مقاومت پذیری[23] را در سطوح مختلفی تامین کرده و ارتباطات از طریق سرورهای مرکزی مبادله نمی‌شوند و یا ذخیره نمی‌گردند. برای پیوستن به شبکه Tox کافی است که به گره‌های شناخته شده‌ای موسوم به bootstrap متصل شویم. هر کسی می‌تواند یک گره bootstrap را راه‌اندازی کند و نیاز نیست که کاربران به آن اعتماد داشته باشند. Tox تلاش می‌کند تا ارتباطات را در شرایطی که گره‌ها پشت دیواره آتش و یا NAT پنهان شده باشند، نیز برقرار کند. این شبکه در برابر حملات پایه DoS مقاومت می‌کند. وجود زمان‌های انقضاءکوتاه، باعث پویایی شبکه و مقاومت پذیری در برابر تلاش‌های مخرب[24] می‌شود.

Tox تلاش می‌کند که کاربر پسند بوده و نیازمند حداقل پیکربندی باشد. این مساله کمک می‌کند که تامین و دسترسی به امنیت، برای کاربران معمولی ساده و دست‌یافتی باشد. این نرم افزار به‌راحتی قابل پیکربندی با نرم افزارهایی مانند Tor است که ناشناس ماندن در فضای مجازی را تامین می‌کنند.

---

[20] Secret key
[21] End-to-end encryption
[22] Onion routing protocol
[23] Resilience
[24] Poisoning attempts

۴-۲ داده‌ساختارهای Tox

داده‌ساختارهای Tox شامل موارد زیر است:

- **Node Info**: داده‌ساختاری است که شامل پروتکل انتقال، آدرس سوکت و کلید عمومی است. پروتکل انتقال می‌تواند TCP یا UDP باشد. آدرس سوکت متشکل از آدرس میزبان و شماره پورت است. آدرس میزبان می‌تواند مبتنی بر IPv4 یا IPv6 باشد.
- **Client List**: برای مدیریت و سازماندهی گره‌ها و عملیات مختلف، نیازمند تعریف مفهوم فاصله بین گره‌ها هستیم. در این شبکه فاصله بین دو گره برابر مقدار عددی حاصل از XOR کردن کلید عمومی آنهاست. با استفاده از معیار فاصله، مفهومی به نام Client List تعریف می‌شود. یک Client List با اندازه حداکثر k، عبارت است از مجموعه‌ای مرتبط از حداکثر k گره که نزدیک‌ترین فاصله را به یک کلید مشخص موسوم به کلید پایه[25] دارند.
- **K-Buckets**: برای ذخیره موثر گره‌های نزدیک به هم داده ساختاری به نام K-Buckets تعریف می‌شود که یک شاخص موسوم به Bucket Index را به یک Client List با اندازه حداکثر k، نگاشت می‌کند. این شاخص عددی است که مقدار آن حداقل ۰ و حداکثر ۲۵۵ است. در اینجا به Client List، k-bucket نیز گفته می‌شود که اندازه آن برابر k است. هرچه k بزرگتر باشد، سرعت پیدا کردن گره‌های نظیر بیشتر است. اگر یک گره، داخل یک k-bucket با شاخص n قرار داشته باشد، آنگاه این k-bucket شامل گره‌هایی است که فاصله آنها با کلید پایه در محدوده $[2^n, 2^{n+1} - 1]$ قرار دارد.
- **حالت گره[26]**: حالت هر گره DHT شامل اطلاعات زیر است :
  ۱. زوج کلیدهای DHT که شامل کلید عمومی و کلید رمز هستند و از آن برای ارتباط با سایر گره‌ها از آن استفاده می‌کند.
  ۲. لیستی از گره‌های نزدیک[27] که آن را با داده ساختار K-Buckets مدل می‌کند که در آن کلید عمومی DHT، همان کلید پایه است و به ازای هر شاخص دارای یک Client List است.
  ۳. لیستی از کلیدهای عمومی‌جستجو شده[28] که متناظر با هر کدام آنها، یک مدخل جستجو ایجاد می‌شود که در واقع یک Client List است که کلید پایه آن، همان کلید عمومی گره‌ای است که آن را جستجو کرده‌ایم.

حالت اولیه هر گره شامل زوج کلید DHT است و لیست گره‌های نزدیک و لیست گره‌های جستجو شده خالی هستند.

- **Protocol Packet**: Protocol Packet سطح بالاترین بسته در پروتکل Tox است که برای بسته‌بندی سایر بسته‌ها، مورد استفاده قرار می‌گیرد. این بسته شامل دو بخش است :
  - Packet Kind: شناسه‌ای است که نوع بسته را مشخص می‌کند
  - Payload: در برگیرنده بسته‌ای است که نوع آن از طریق شناسه مشخص شده است.

---

[25] Base key
[26] Node state
[27] DHT Close List
[28] DHT Search List

برای انتقال این بسته در شبکه می‌توان از پروتکل‌های UDP یا TCP استفاده کرد. Protocol Packet رمزنگاری نمی‌شود اما محتویات Payload می‌تواند رمزنگاری شوند. برخی از شناسه‌هایی که نوع بسته را مشخص می‌کنند در جدول زیر نشان داده شده است.

Byte value	Packet Kind
0x00	Ping Request
0x01	Ping Response
0x02	Nodes Request
0x04	Nodes Response

- **DHT Packet**: DHT Packet شامل اطلاعات زیر است:
  ۱. کلید عمومی فرستنده
  ۲. یک مقداری تصادفی Nonce جهت رمزنگاری
  ۳. Payload که با استفاده از کلید رمز فرستنده، کلید عمومی گیرنده و مقدار nonce، رمزنگاری شده است

  DHT Packet در داخل Protocol Packet و در قالب انواع مختلف آن ارسال می‌شود.

- **DHT Request Packet**: این بسته برای هدایت داده‌های رمزنگاری شده از فرستنده به گیرنده، از طریق گره‌های ثالث، مورد استفاده قرار می‌گیرد و در قالب بدنه[29] Protocol Packet و متناظر با شناسه‌ای[30] که نوع بسته را نشان می‌دهد، ارسال می‌شود. این بسته حاوی اطلاعات زیر است:
  ۱. کلید عمومی گیرنده
  ۲. DHT Packet که باید توسط گیرنده دریافت شود.

  هنگامی که یک گره درخواستی را دریافت می‌کند، ابتدا مقدار کلید عمومی گیرنده را با مقدار کلید عمومی خودش مقایسه می‌کند. اگر این دو مقدار، یکی بودند کار رمزگشایی و اداره کردن درخواست را انجام می‌دهد در غیر اینصورت، بررسی می‌کند که آیا کلید عمومی گیرنده، در میان لیست گره‌های نزدیک به او قرار دارد یا نه. در صورت وجود، این بسته بدون تغییر به آن گره ارسال می‌شود و در غیر اینصورت دور انداخته می‌شود.

۴-۳ ماژول‌های تشکیل‌دهنده Tox

۴-۳-۱ ماژول Crypto

این ماژول شامل توابع و نوع داده‌هایی است که مرتبط با رمزنگاری است. در پروتکل Tox، بسته‌ها با استفاده از کلید عمومی گیرنده و کلید رمز فرستنده کد می‌شود.گیرنده نیز با دریافت بسته، آن را با کلید رمز خود و کلید عمومی فرستنده، رمزگشایی

---

[29] Payload
[30] Packet Kind

می‌کند. پروتکل Tox بین دو نوع متن ساده و متن رمزگذاری شده تفاوت قائل است. متن رمزگذاری شده می‌تواند از طریق کانال غیر امن ارسال شود. متن‌های ساده نیز به دو دسته متن حساس و غیر حساس تقسیم می‌شود. متون حساس باید حتما قبل از ارسال از طریق کانال‌های غیر امن، رمزگذاری شوند.

۴-۳-۲ ماژول DHT

DHT مجموعه‌ای خود-سازماندهی‌شونده از گره‌هاست که در شبکه Tox با یکدیگر مشارکت و همکاری می‌کنند. وظیفه این ماژول پیداکردن آدرس IP و شماره پورت گره‌ها و همچنین برقراری یک مسیر مستقیم به گره‌ها است. DHT برای این کار از پروتکل UDP استفاده می‌کند. هر گره در DHT دارای یک زوج کلید موقتی است که به آن زوج کلید DHT می‌گویند و شامل کلید عمومی DHT[31] و کلید رمز DHT[32] می‌باشد. کلید عمومی DHT به عنوان آدرس گره عمل می‌کند. هرگاه یک نمونه از Tox بسته شود و یا راه‌اندازی مجدد گردد، زوج کلید DHT مجددا ایجاد می‌شوند[33]. برای پیدا کردن کلید عمومی DHT دوستان از ماژول Onion استفاده می‌شود. هنگامی که کلید عمومی DHT یک دوست را بدانیم، می‌توانیم با کمک DHT او را پیداکرده و با کمک پروتکل UDP به طور مستقیم به او متصل شویم. البته در DHT ؛ گره‌هایی وجود دارد که کلید عمومی آنها پایدار است و بر اثر راه‌اندازی مجدد، تغییر نمی‌کنند که به آنها DHT Bootstap Node گفته می‌شوید. در DHT سرویس‌هایی وجود داردکه به کمک آنها می‌توانیم زنده بودن یک گره را بررسی کنیم و یا با کمک آن، یک گره DHT دیگر را جستجو کنیم.

۴-۳-۳ ماژول TCP Server

هدف TCP Server در Tox این است که مانند یک رله TCP برای گره‌هایی که نمی‌توانند به طور مستقیم به یکدیگر متصل شوند، عمل کند. برای اتصال به TCP Server از ماژول TCP Client استفاده می‌شود و اتصال بین آنها رمزگذاری می‌شود تا اطلاعات مربوط به گره‌هایی که به یکدیگر متصل هستند آشکار نگردد و همچنین نتوان پیام‌های ردوبدل شده را دستکاری کرد. TCP Server نقش رله بین دو گره همتا را بازی می‌کند. وقتی TCP Client به سرور متصل می‌شود به او می‌گوید که قصد اتصال به چه مشتری را دارد. پس از برپایی اتصال بین TCP Client و سرور، فرایند Handshaking صورت می‌گیرد. هدف از Handshaking این است که :

۱. سرور باید مطمئن شود که وقتی یک گره خود را با یک کلید عمومی معرفی می‌کند، کلید خصوصی آن را نیـز دارد. به عبارت دیگر، همان کسی است که ادعای آن را می‌کند.
۲. باید یک اتصال امن که دارای forward Secrecy است را پایه‌گذاری کنیم.
۳. مانع از حملات شویم.

این کار باعث می‌شود که اگر مشتری، کلید خصوصی مرتبط با کلید عمومی اعلامی را نداشته باشد، نمی‌تواند فرایند Handshaking را انجام دهد. همچنین از حملات مختلفی جلوگیری می‌شود:

---

[31] DHT Public Key
[32] DHT Secret Key
[33] Renewed

۱. حمله‌کننده نمی‌تواند اطلاعات بسته Handshaking را تغییر دهد.
۲. نمی‌تواند بسته Handshaking ارسال شده از سمت مشتری به سرور را گرفته و آن را بعدا برای سرور ارسال کند.
۳. نمی‌تواند پاسخ سرور را گرفته و هنگامی که مشتری در آینده قصد اتصال بـه سـرور را داشـته باشـد، آن را بـرای او ارسال کند.
۴. نمی‌تواند خود را جای مشتری، به سرور معرفی کند.
۵. نمی‌تواند خود را جای سرور، به مشتری معرفی کند.

۴-۳-۴ ماژول TCP Client

Client یک مشتری برای TCP Server است و اتصال با او را برقرار و نگهداری می‌کند. وظیفه TCP Client این است که مطمئن باشد بسته‌های مهم، به دلیل شوغ بودن سر سرور، دور ریخته نمی‌شوند و یا اینکه بسته‌های بزرگ به صورت تدریجی به سرور ارسال کند. همچنین می‌توان با کمک آن یک مسیر ارتباطی با دوستانی که به یک TCP Server متصل هستند، باز کرد.

۴-۳-۵ ماژول Onion

هدف از این ماژول این است که مطمئن شویم فقط دوستان یک گره قادر به پیداکردن و اتصال به او هستند و به طور غیرمستقیم ، از یافتن آدرس IP یک گره توسط سایرین جلوگیری کنیم. این ماژول از یافتن کلید عمومی بلندمدت یک گره از روی کلید عمومی DHT کوتاه‌مدت او جلوگیری می‌کند. همچنین از یافتن کلید عمومی DHT کوتاه‌مدت از روی کلید عمومی بلندمدت جلوگیری می‌کند. برای اجتناب از کشف تناظر بین کلید عمومی بلند مدت یک گره با کلید عمومی DHT کوتاه‌مدت آن، گره‌ها از انتشار کلید عمومی بلند مدت خود اجتناب می‌کنند و تنها کلید عمومی DHT کوتاه مدت خود را به سایرین در شبکه می‌دهند. Onion این امکان را به گره‌ها می‌دهد تا کلید عمومی DHT خود را به دوستان خود، که کلید عمومی واقعی آنها را می‌دانند، ارسال کنند.

Onion این امکان را به گره‌ها می‌دهد تا کلید عمومی واقعی خود را از طریق مسیرهای Onion اعلان[34] کنند. یک گره برای اعلان خود در شبکه ابتدا گره‌هایی که کلید عمومی DHT آنها نزدیک به کلید عمومی واقعی خودش است را پیدا می‌کند. سپس کلید عمومی بلندمدت خود را در یک بسته قرار داده و آن را با کلید بلندمدت خصوصی خود رمزگذاری کرده و برای آنها ارسال می‌کند. به این ترتیب، گرهی که خود را به او معرفی کردیم می‌تواند مطمئن باشد که ما واقعا دارای کلید عمومی هستیم. حال اگر دوستان گره X، بخواهند به او پیامی ارسال کنند، ابتدا باید گره‌هایی که X خود را به آنها اعلان کرده است را پیدا کنند. برای این منظور، گره‌های نزدیک به کلید عمومی واقعی X را پیدا می‌کنند و از آنها سوال می‌کنند که آیا گره X را می‌شناسند یا نه. در صورت مثبت بودن پاسخ، می‌توانند از طریق این گره‌ها، پیام خود را به X برسانند. این پیام حاوی کلید عمومی DHT آنها و همچنین اطلاعات رله‌های TCP و برخی از گره‌هایی است که به آنها متصل هستند. این اطلاعات به X این امکان را می‌دهد تا به دوستان خود متصل شود.

---

[34] Announce

۴-۳-۶ پروتکل NetCrypto

پروتکل انتقالی Tox است که از آن برای برقراری اتصال و ارسال امن داده‌ها بین دوستان استفاده می‌شود و امکان رمزگذاری، تحویل مرتب بسته‌ها و یک forward Secrecy کامل را فراهم می‌کند. این ماژول برای این کار از پروتکل UDP استفاده می‌کند اما در شرایطی که دو دوست از طریق رله‌های TCP به هم متصل هستند نیز می‌توان از آن استفاده کرد.

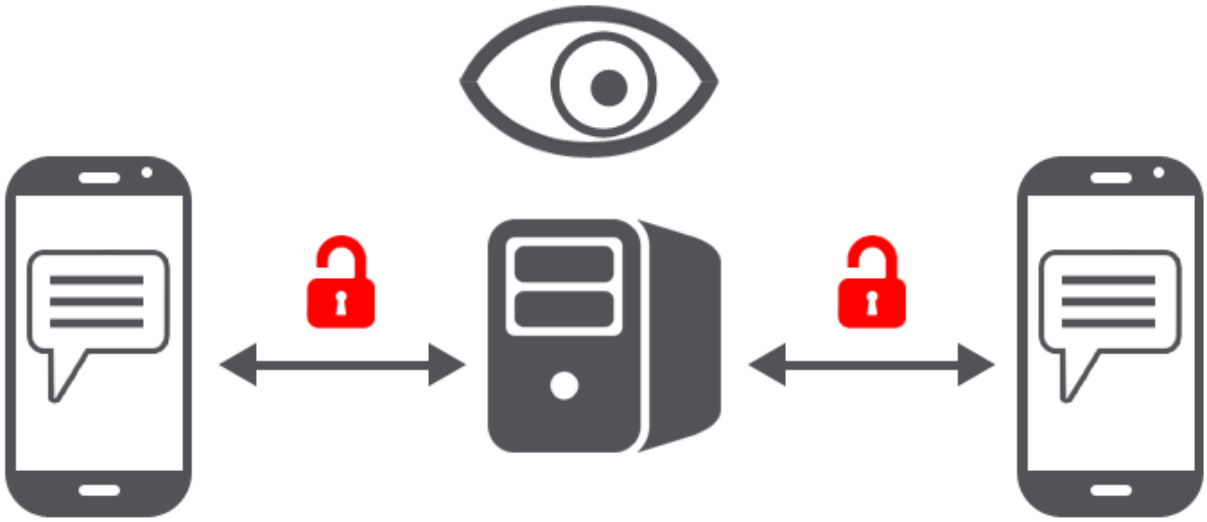
۴-۳-۷ مولفه Friend Connection

ماژولی است که بر روی DHT، Onion و Net Crypto قرار دارد و ارتباط این سه ماژول به یکدیگر را مدیریت می‌کند. این ماژول اتصال به دوستان را برقرار می‌کند و با ارائه یک لایه پیام سطح بالاتر یک واسط ساده برای ارسال و دریافت پیام و افزودن و یا حذف کردن دوستان و اطلاعات از آنلاین بودن یا آفلاین بودن آنها فراهم می‌کند.

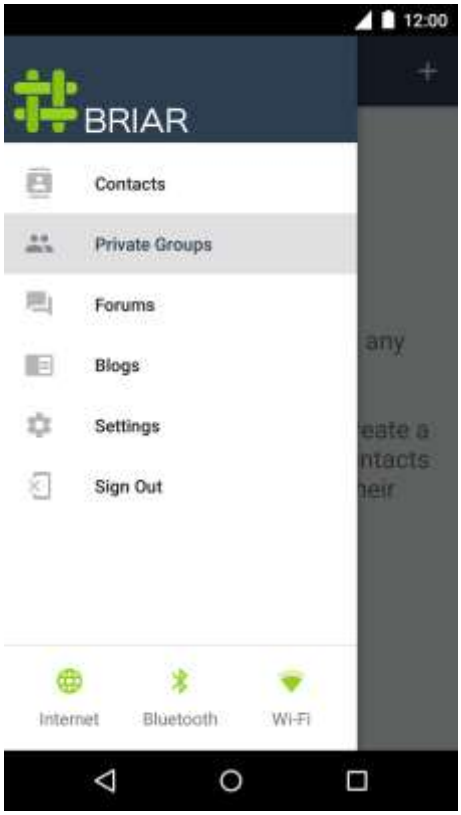
فصل ۵ نرم‌افزار پیام‌رسان Briar

Briar یک نرم‌افزار پیام‌رسان است که برای فعالین اجتماعی، روزنامه‌نگاران و همه کسانی که به دنبال یک راه‌حل ساده، امن و مستحکم برای ارتباط هستند طراحی شده است. برخلاف رویکردهای معمول در نرم‌افزارهای پیام‌رسان این سیستم متکی به سرور مرکزی نیست و همگام‌سازی پیام‌ها به صورت مستقیم بین دستگاه‌های کاربران صورت می‌گیرد. اگر اینترنت دچار اختلال شود می‌تواند همگام‌سازی را از طریق بلوتوث یا شبکه Wi-Fi انجام دهد و جریان داده را در شرایط بحرانی منتقل کند. در صورت برقراری اتصال به اینترنت، می‌تواند با Tor همگام شده تا از کاربران و ارتباطات بین آنها محافظت کند.

Briar از اتصال مستقیم و رمزنگاری شده بین کاربران برای جلوگیری از جاسوسی استفاده می‌کند در حالیکه پیام‌رسان‌هایی که متکی به سرور مرکزی هستند پیام‌ها و ارتباطات را می‌توانند در معرض جاسوسی قرار دهند.



Briar امکان پیام‌رسانی خصوصی، انجمن‌ها و وبلاگ‌های عمومی را ارائه می‌کند و از آنها در برابر جاسوسی و سانسور محافظت می‌کند. این نرم‌افزار با استفاده از شبکه Tor مانع از استراق سمع و مشخص شدن اینکه چه افرادی با یکدیگر در حال مکالمه هستند می‌شود. لیست تماس‌های هر کاربر رمزنگاری شده و در دستگاه خود او ذخیره می‌شود.



با استفاده از رمزنگاری سرتاسری[35] از دستکاری و استراق سمع داده‌ها جلوگیری می‌شود. همچنین این مساله مانع از فیلترینگ بر اساس کلیدواژه می‌شود و با توجه به ماهیت غیر متمرکز این سیستم، امکان بلاک کردن آن نیز وجود ندارد. هر کاربری که عضو انجمن است، یک نسخه کپی از داده‌ها را دارد و بنابراین یک نقطه مشخص برای پاک کردن آن و یا از دسترس خارج کردن آن از طریق حملات DoS وجود ندارد.

۵-۱ نحوه عملکرد

برای آشنایی با نحوه کارکرد این سیستم، پروتکل‌های اصلی و سطح بالای آنر در ادامه شرح می‌دهیم.

۵-۱-۱ پروتکل BQP[36]

از این پروتکل برای پایه‌گذاری یک کلید رمز بین هر دو دستگاه استفاده می‌شود و درواقع یک پروتکل توافق روی کلید است. این دستگاه‌ها باید در همسایگی هم قرار داشته باشند و امکان ارتباط و تعامل از طریق بستر انتقالی دو طرفه بردکوتاه[37] را باید داشته باشند اما لزوما این لایه انتقال دارای ویژگی‌های امنیتی نیست. دستگاه‌ها می‌توانند کد QR یکدیگر را اسکن کرده و با کمک اطلاعات موجود در آن، یک اتصال غیر امن با یکدیگر برقرار کنند و کلید عمومی خود را مبادله کنند. هر کد QR

[35] End-to-end encryption
[36] Bramble QR code Protocol
[37] Short-range bidirectional transport

شامل یک کلید عمومی کوتاه مدت و اطلاعاتی در مورد نحوه اتصال با سایر دستگاه‌ها است. اجرای این پروتکل در قالب چهار فاز آماده سازی، برقراری اتصال، توافق روی کلید، استخراج شاه‌کلید انجام می‌شود.

هریک از دستگاه‌ها با کمک کلید خصوصی خود و کلید عمومی دریافتی از دستگاه دیگر، یک کلید رمز اشتراکی ایجاد می‌کنند که می‌توان از روی آن یک شاه‌کلید ساخت و آن را برای انجام تبادلات امن بر بستر انتقالی به کار گرفت. فرض بر این است که تبادل اولیه کد QR در برابر حمله man-in-the-middle محفوظ است زیرا کاربر می‌بیند که در حال اسکن کردن کدام دستگاه است و در نتیجه مطمئن است که کد اسکن شده مربوط به فردی است که قصد تبادل کلید را با او دارد. در تبادلات بعدی کلید عمومی بر بسته اتصال غیر امن نیز این حمله از طریق مقایسه کلیدها با اطلاعات موجود در QR که روی آن توافق شده است قابل کشف است.

۲-۱-۵ پروتکل BSP[38]

این پروتکل به منظور همگام سازی داده‌ها در لایه کاربرد[39] به کار می‌رود که برای شبکه‌های مقاوم در برابر تاخیر[40] مناسب است. این پروتکل، داده‌ها را در میان مجموعه‌ای پویا از دستگاه‌ها همگام می‌کند. هر کدام از این دستگاه‌ها با مجموعه‌ای پویا از همتایان[41] خود در ارتباط هستند. داده‌ها در قالب گروه‌ها سازماندهی می‌شوند و هر گروه حوزه همگام سازی مستقلی دارد که شامل یک گراف پیام[42] است که از روی پیام‌های غیر قابل تغییر ساخته می‌شود.

---

[38] Bramble Synchronisation Protocol
[39] Application layer
[40] Delay-tolerant networks (DTN)
[41] Peers
[42] Message graph

اگر یک وسیله، در همگام‌سازی پیام‌های یک گروه مشارکت کند اصطلاحا عضو گروه است. اگر دو گره همتا، پیام‌های گروه را با یکدیگر همگام‌سازی کنند اصطلاحا گفته می‌شود که گروه را بین یکدیگر به اشتراک گذاشته اند. لزومی ندارد همتایانی که عضو یک گروه هستند، آن را با یکدیگر به اشتراک بگذارند. هر یک از اعضای گروه، یک بخشی از گراف پیام را در قالب یک کپی جزئی[43] ذخیره سازی می‌کند. هر پیام در این کپی جزئی از گراف می‌تواند به اشتراک گذاشته شده و یا پاک شود. اگر پیام به اشتراک گذاشته شود، دستگاه آن پیام را با همه همتایانی که گروه را به اشتراک گذاشته اند، همگام خواهد کرد. اگر پیام پاک شود، دستگاه کپی آن پیام را حذف می‌کند اما اطلاعاتی در مورد موقعیت آن در گراف پیام نگهداری می‌کند.

هر گروه متعلق به یک مشتری[44] است که در واقع همان برنامه‌ای است که از این پروتکل برای همگام‌سازی داده‌ها استفاده می‌کند. مشتری است که تصمیم می‌گیرد که گروه با چه همتایانی به اشتراک گذاشته شود و چه چیزی یک پیام معتبر را می‌سازد، چه پیامی باید به اشتراک گذاشته شود و چه پیامی باید حذف شود و وظیفه پروتکل همگام‌سازی این است که این تصمیمات را از طرف مشتری پیاده‌سازی کند.

این پروتکل، نیازمند پروتکل امنیتی لایه انتقال است که می‌تواند داده‌ها را به بهترین وجه از یک دستگاه به دستگاه دیگر تحویل دهد. در حالیکه داده‌ها در معرض انتقال با تاخیر، گم شدن، جابجایی و یا تکرار شدن هستند، این وظیفه پروتکل امنیتی لایه انتقال است که محرمانگی، یکپارچه‌گی، اعتبارسنجی و forward secracy داده‌ها را تامین کنند. در این پروتکل، مشتری دارای شناسه یکتاست تا از برخورد بین گروه‌ها و پیام‌های آنها اجتناب شود. شناسه هر گروه یکتا است و از روی شناسه مشتری و توصیف گروه ایجاد می‌شود. پیامها نیز دارای برچسب زمانی هستند و شناسه آنها باکمک شناسه گروه و برچسب زمانی تولید می‌شود.

فرایند همگام سازی از طریق مبادله رکوردهای اطلاعاتی بین فرستنده و گیرنده صورت می‌گیرد که می‌تواند انواع مختلفی داشته باشد. به عنوان مثال، فرستنده می‌تواند شناسه پیام‌هایی که در اختیار دارد را برای گیرنده ارسال کند و یا اینکه با ارسال شناسه پیام‌ها، از گیرنده بخواهد که این پیام‌ها را در اختیار او قرار دهد. همچنین فرستنده می‌تواند شناسه پیام‌هایی که می‌تواند با گیرنده به اشتراک بگذارد را برای او ارسال کند. در بدیهی‌ترین حالت، رکورد اطلاعاتی حاوی خود پیام است. فرایند همگام سازی می‌تواند به دو صورت انجام شود. یا به صورت تعاملی[45] و یا به صورت دسته‌ای[46]. همگام سازی به صورت تعاملی نیازمند پهنای باند کمتری است اما نیازمند دو رفت و برگشت[47] است.

در حالت تعاملی، پیام‌ها قبل از ارسال، پیشنهاد می‌شوند و دستگاه به این صورت عمل می‌کند که برای هر کدام از پیام‌هایی که دریافت کرده است و یا به او پیشنهاد شده است ولی تاکنون برای آن تائیدیه ارسال نشده است، تائیدیه ارسال می‌کند. همچنین برای پیام‌هایی که به او پیشنهاد شده است و او آنها را ندارد و تاکنون نیز برای آن درخواستی را ارسال نکرده است، درخواست ارسال می‌کند. در صورتیکه دستگاه، برای یکی از پیام‌هایی که به اشتراک‌گذاشته است درخواستی را از یکی از همتایان دریافت کند، آن را ارسال می‌کند. همچنین پیام‌هایی را که به اشتراک گذاشته است اما از داشتن آنها توسط همتایان خود مطمئن نیست را به آنها پیشنهاد می‌کند. در حالت دسته‌ای، پیام‌ها بدون اینکه پیشنهاد شوند، ارسال می‌گردند. در

---

[43] Partial copy
[44] Client
[45] Interactive Mode
[46] Batch mode
[47] Round-trips

چنین حالتی، پیام‌هایی که به اشتراک گذاشته شده است و برای آن درخواستی دریافت می‌شود را ارسال می‌کند. همچنین اگر از داشتن این پیام‌های اشتراکی توسط سایر همتایان خود مطمئن نباشد، آنها را ارسال می‌کند.

گراف پیام، یک گراف جهت‌دار است و تشکیل یک DAG را می‌دهدکه در آن هر پیام به آنچه وابسته است اشاره می‌کند. در واقع لیست وابستگی‌های یک پیام، موقعیت آن را در گراف مشخص می‌کند. زمانی یک پیام را می‌توانیم تحویل مشتری دهیم که همه وابستگی‌های آن را قبلا تحویل داده باشیم. در واقع تلاش می‌شود تا سازگاری علی تامین شود.

۳-۱-۵ پروتکل BTP[48]

BTP یک پروتکل امن لایه انتقال است که برای شبکه‌های همپوشان مقاوم در برابر تاخیر[49] مناسب است و یک کانال امن بین دو نقطه پایانی دستگاه‌ها ایجاد می‌کند تا از محرمانگی[50]، یکپارچگی[51]، اعتبار و صحت[52] و رمز عبور روبه جلو[53] ر ارتباطات شکل گرفته روی بازه وسیعی از زیرساخت و بستر انتقال، اطمینان حاصل کنیم. اصلی‌ترین مولفه BTP، پروتکل مدیریت کلید وابسته به زمان و همچنین شیوه[54] انتقال امن جویباری از داده‌هاست. این پروتکل با تامین ویژگی‌های امنیتی سازگار با طیف وسیعی از شیوه‌های انتقال، توسعه شبکه‌های همپوشان مقاوم در برابر تاخیر و سانسور[55] را تسهیل کرده است.

BTP تلاش نمی‌کند تا هویت طرف‌هایی ارتباط و یا این واقعیت که آنها در حال ارتباط با هم هستند را پنهان کند و ناشناس بودن[56]، غیر قابل اتصال بودن[57] و غیر قابل مشاهده بودن[58] را تامین نمی‌کند. اگر چنین ویژگی‌هایی مورد نیاز باشد، BTP می‌تواند از سیستم‌هایی نظیر Tor یا Mixminion به عنوان زیربنای انتقال خود استفاده کند. رمز عبور رو به جلو از طریق برپایی اولیه رمز مشترک[59] برای هر زوج از نقاط پایانی دستگاه‌ها و استفاده از یک تابع یک‌طرفه استخراج کلید جهت تولید دنباله‌ای از کلیدهای موقتی از روی رمز مشترک صورت می‌گیرد. هر گاه هر دو دستگاه یک کلید را پاک کنند دیگر نمی‌توان آن را استخراج کرد.

از آنجایی که هدف BTP این است که در سیستم‌های مقاوم در برابر جاسوسی و سانسور مورد استفاده قرار گیرد بنابراین شرایط زیر را باید فرض کرد و نیازمندیهای طراحی را متناسب با آن تنظیم کرد. امکان مشاهده، مسدود کردن، پاسخ دادن و تغییر دادن ترافیک در زیرساخت زیرین انتقال توسط افراد و سازمان‌های بدخواه وجود دارد. آنها می‌توانند با کمک یک پروتکل سطح بالاتر، داده‌های نوشته شده در لایه BTP را انتخاب کنند. آنها این توانایی را به صورت محدود دارند که نقاط پایانی دستگاه را در معرض خطر قرار دهند و در صورتیکه موفق به اینکار شوند آنگاه می‌توانند به همه اطلاعات ذخیره شده در

---

[48] Bramble Transport Protocol
[49] Delay-tolerant network
[50] confidentiality
[51] Integrity
[52] Authenticity
[53] Forward Secrecy
[54] Wire protocol
[55] Censorship-resistance
[56] Anonymity
[57] Unlikability
[58] Unobservability
[59] Shared secret

حافظه و فضای ذخیره سازی دسترسی داشته باشند. بنابراین حداقل نیازمندیهای امنیتی BTP در مواجهه با این فرضیات این است که :

۱. در صورت دسترسی به داده‌هایی که در بستر BTP منتقل می‌شود، نتوان از آن چیزی فراگرفت.
۲. در صورت دستکاری جویبار داده‌ها این مساله توسط گیرنده نقطه پایان قابل کشف است
۳. نباید این امکان وجود داشته باشد که داده‌ای توسط یک شخص ثالث، به یکی از نقاط پایانی ارسال شود و اینگونه وانمود گردد که این داده از سمت دیگر ارتباط ارسال شده است.
۴. اگر در آینده، یکی یا هر دو نقطه پایانی توسط افراد مخرب مورد نفوذ قرار گرفت، نمی‌توانند از آنچه که قبلا ارسال شده است اطلاع حاصل کنند.