
# تونل SSH برای اتصال به کامپیوترهای راه دور

**محسن سعید زاده ***[1]

## چکیده

هدف از این تحقیق تونل SSH برای اتصال به کامپیوترهای راه دور می باشد، Secure Shell (**SSH**)یک روش امن برای دسترسی از **راه دور به** سرورها ارائه می‌دهد، اصولاً در محیطهای Unix و Linux استفاده می‌شود .این امکان را فراهم می‌کند که دسترسی خط فرمان رمزنگاری شده فراهم شود که اجازه انتقال امن فایل، اجرای دستور از **راه دور** و تونلینگ را می‌دهدما نرم افزاری را برای ایجاد اتصالات پوسته ایمن (خط فرمان) به رایانه های راه دور برای انجام تعمیرات، پیکربندی یا اشکال زدایی ارائه می دهیم، حتی زمانی که رایانه های راه دور به اینترنت متصل هستند به طوری که اتصالات ورودی مجاز نیستند. نصب و راه اندازی این نرم افزار آسان است، وابستگی های کمی دارد، ضمانت های امنیتی بسیار مفیدی را ارائه می دهد و برای تعداد زیادی رایانه از راه دور که احتمالاً به گروه های جدا شده تقسیم می شوند، مقیاس می شود. این نرم افزار به یک یا دو سرویس ابری ارزان قیمت متکی است. نسخه اولیه این سیستم به مدت ۹ سال برای دسترسی به رایانه‌های از راه دور جغرافیایی که به عنوان حسگر در سیستم‌های ردیابی حیات وحش عمل می‌کنند، استفاده شده است.

**کلمات کلیدی:** دسترسی از راه دور، پوسته، ایمن، تونل های SSH،

[1] مدرس دانشگاه، کامپیوتر، دانشگاه فنی حرفه ای پسران، یاسوج، ایران

Mohsen.saeedzadeh2024@gmail.com

## مقدمه

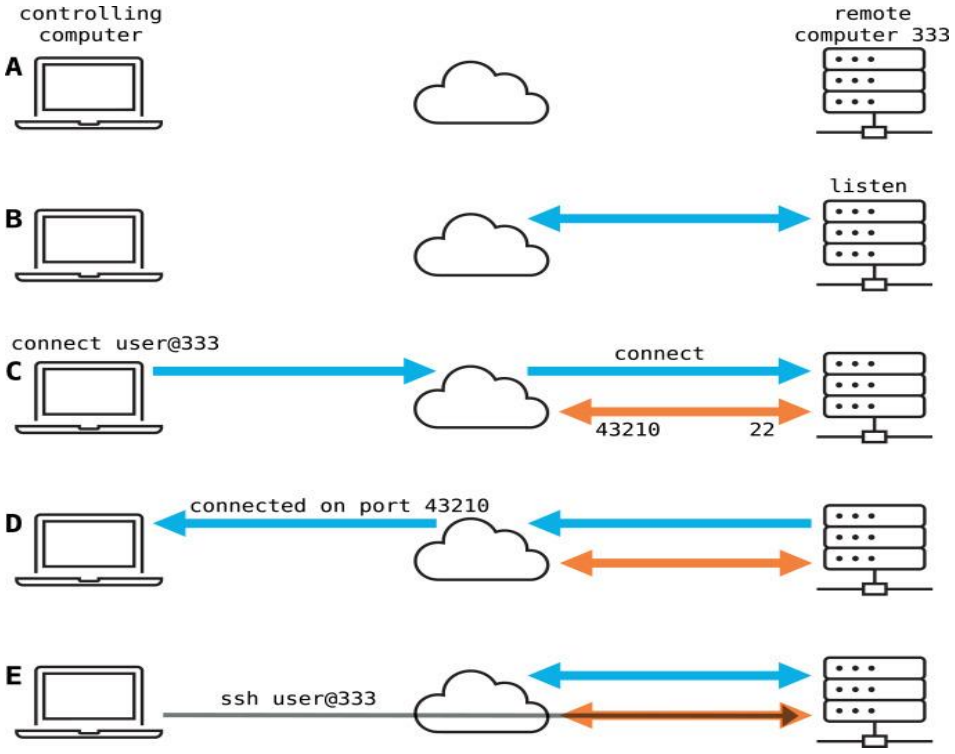
رایانه‌های مقرون‌به‌صرفه و مودم‌های سلولی مقرون‌به‌صرفه، دانشمندان را قادر می‌سازد تا رایانه‌های متصل به اینترنت از راه دور را برای جمع‌آوری داده‌های حسگر و کنترل آزمایش‌ها مستقر کنند (به عنوان مثال، [۱] را ببینید). رایانه‌های راه دور که سیستم‌عامل‌های استانداردی مانند لینوکس یا ویندوز را اجرا می‌کنند، به دانشمندان این امکان را می‌دهند که از طیف وسیعی از نرم‌افزارها در این رایانه‌ها استفاده کنند و در صورت نیاز نرم‌افزار و داده‌های پیکربندی را به‌راحتی تعمیر یا به‌روزرسانی کنند. این اقدامات تعمیر و نگهداری نیاز دارد که مدیر سیستم یک اتصال به یک برنامه پوسته (خط فرمان) در حال اجرا بر روی رایانه راه دور، معمولاً با استفاده از پروتکل پوسته ایمن (SSH)باز کند [۲]. با این حال، اتصال اینترنتی این کامپیوترهای راه دور معمولاً اجازه اتصالات ورودی SSH) یا هر نوع دیگر) را نمی دهد. این ممکن است به دلیل یک دیوار آتش یا مکانیسم ترجمه آدرس شبکه باشد که ارائه دهنده اینترنت برای محافظت از مشتریان خود یا رفع نیاز به بسیاری از آدرس های IP نسخه ۴ با آدرس دهی جهانی استفاده می کند. [۳]

نرم افزار ما به مدیر سیستم اجازه می دهد تا از رایانه لینوکس یا ویندوز خود که ما به عنوان رایانه کنترل کننده از آن یاد می کنیم، به تعداد زیادی از رایانه های راه دور دارای لینوکس که از اتصالات شبکه ورودی پشتیبانی نمی کنند و به ویژه اتصالات SSH ورودی را پشتیبانی نمی کنند، متصل شود. این نرم افزار به سه سرویس متکی است که معمولاً در فضای ابری ارائه می شوند. اولین مورد، یک کامپیوتر با یک IP یا آدرس DNS ثابت است که به عنوان یک پروکسی استفاده می شود، همچنین به عنوان میزبان پرش در اسناد SSH شناخته می شود. برای ایجاد اتصال از یک کنترل کننده به یک کامپیوتر راه دور، هر دو یک اتصال به اصطلاح SSH تونلی [۴] را به پراکسی باز کنید. تونل ها یک شبکه مجازی را فراهم می کنند که از طریق آن کامپیوتر کنترل کننده یک اتصال SSH با کامپیوتر راه دور برقرار می کند. تونل ها به صورت ایمن و خودکار ایجاد می شوند. کاربر فقط باید بگوید که می خواهد به کدام کامپیوتر راه دور متصل شود. سرویس دوم یک حساب GitHub است که به عنوان مخزن کلیدهای عمومی SSHو گواهی های X.509 استفاده می شود GitHub .یک سیستم مبتنی بر وب محبوب برای مدیریت کد منبع و فایل های دیگر است. سومین سرویس که اختیاری است، یک کارگزار انتشار–اشتراک خدمات وب آمازون (AWS)است. این سرویس پروتکل عمومی MQTT را به صورت ایمن و مقیاس پذیر پیاده سازی می کند [۴].

اگر از کارگزار استفاده نمی شود، تونل های کامپیوترهای راه دور به میزبان پرش باید همیشه باز باشند. این تعداد کامپیوترهای راه دور را به ده ها، شاید چند صد نفر محدود می کند. یک کارگزار به همان میزبان پرش اجازه می دهد تا به تعداد بسیار بیشتری از رایانه های راه دور سرویس دهد، با اجازه دادن به رایانه کنترل کننده برای ارسال دستورات به رایانه راه دور برای راه اندازی یا خراب کردن تونل به میزبان پرش. ما به اولین پیکربندی تونل ها به عنوان پیکربندی همیشه روشن و به پیکربندی دوم به عنوان پیکربندی درخواستی اشاره می کنیم.

یک نمونه واحد از هر یک از این سه منبع که ما از آنها به عنوان یک قلمرو یاد می‌کنیم، می‌تواند به چندین گروه از رایانه‌های راه دور خدمت کند که ما به آنها سیستم می‌گوییم. هر سیستم معمولاً توسط یک مدیر (یا چندین مدیر) متفاوت مدیریت می شود. نرم افزار ما جداسازی بین سیستم ها را فراهم می کند. مؤلفه‌های نرم‌افزاری زیربنایی که نرم‌افزار ما استفاده می‌کند، قوی و منبع باز هستند و سرویس‌های ابری که نرم‌افزار استفاده می‌کند، قوی و رایگان یا ارزان هستند [۵].

هنگامی که نرم افزار راه اندازی شد، همانطور که در شکل ۱ نشان داده شده است عمل می کند. (B) کامپیوتر راه دور یک اتصال انتشار–اشتراک MQTT به یک سرویس ابری ایجاد کرد و به دستورات گوش می دهد. (C) هنگامی که کاربر می خواهد به رایانه راه دور متصل شود، فرمان اتصال را صادر می کند که نام کاربری و نام رایانه را مشخص می کند، ۳۳۳ در شکل. رایانه کنترل کننده یک فرمان اتصال را در مورد موضوعی که رایانه راه دور به آن گوش می دهد منتشر می کند و منتظر پاسخ می ماند. کامپیوتر راه دور یک تونل به اصطلاح معکوس SSH را به یک میزبان پرش (رایانه ای که یک سرور SSH را اجرا می کند، معمولاً یک رایانه مجازی در ابر) ایجاد می کند. (د) شماره پورت روی میزبان پرش را که تونل به آن متصل است (۴۳۲۱۰ در شکل) از طریق اتصال MQTT به کامپیوتر کنترل کننده ارسال می کند. (E) سپس کامپیوتر کنترل کننده یک تونل SSH رو به جلو به میزبان پراکسی و از طریق آن یک اتصال SSH به کامپیوتر راه دور ایجاد می کند.




شکل ۱. مراحل اتصال به یک کامپیوتر راه دور. برای جزئیات به متن مراجعه کنید. [۶]

## ۲- کاربر رابط

رابط کاربری نرم افزار یک اسکریپت تک bash یا cmd.exe به نام tunnel است که یک فرمان واحد را به عنوان آرگومان می گیرد، اغلب با آرگومان. برخی از دستورات از مدیران حوزه پشتیبانی می کنند که نرم افزار را برای یک یا چند سیستم راه اندازی می کنند. سایر دستورات از مدیران سیستم پشتیبانی می کنند که مسئول پیکربندی رایانه های کنترل و راه دور هستند. یک فرمان توسط مدیران سیستم یا کاربران نهایی برای ایجاد یک اتصال SSHبه یک کامپیوتر راه دور استفاده می شود. همچنین یک فرمان کاربر برای از بین بردن یک تونل بین رایانه راه دور و پراکسی وجود دارد، اما معمولاً از آن استفاده نمی‌شود زیرا در حوزه‌های درخواستی، پس از بستن اتصال SSHتوسط کاربر، تونل به طور خودکار از بین می‌رود [۷].

*نسخه ویندوز اسکریپت فقط از دستوراتی پشتیبانی می کند که در رایانه های کنترلی مورد نیاز است.*

برخی از دستورات یک برنامه جاوا را فراخوانی می کنند که بخشی از نرم افزار است. برنامه جاوا تمام ارتباطات انتشار–اشتراک با کارگزار را در صورت استفاده از آن انجام می دهد. همچنین اتصالات SSH تونلی و کنترل اتصالات SSH از راه دور را در حوزه های درخواستی ایجاد می کند. این کار با فراخوانی برنامه باینری سیستم های ssh با آرگومان های مناسب انجام می شود. برنامه جاوا همچنین کلیدهای عمومی، درخواست های امضای گواهی و گواهینامه ها را به شرح گواهینامه های X.509 از و از مخزن GitHub آپلود و دانلود می کند. این فایل‌ها به‌عنوان مشکلات GitHub آپلود و دانلود می‌شوند، نه به‌عنوان فایل‌های معمولی تا هزینه‌های مدیریت فایل که معمولاً با مخازن GitHub مرتبط است کاهش یابد. استفاده از GitHub به مدیران سیستم اجازه می دهد تا تصمیم بگیرند که به کدام کاربران GitHub برای آپلود کلیدهای عمومی اعتماد دارند. یکی دیگر از وظایف برنامه جاوا این است که کلیدهای عمومی را به فایل های SSH authorized_keys هم در رایانه های راه دور و هم در میزبان پرش اضافه کند. در نهایت، برنامه جاوا همچنین از AWS می‌خواهد تا گواهی‌هایی را امضا کند که به رایانه‌های کنترلی و راه دور اجازه می‌دهد به IoT Core متصل شوند. همچنین می‌تواند این گواهی‌ها را فهرست، حذف و دانلود کند.

## ۳- بهترین شیوه های امنیتی

*طراحی نرم افزار از روش هایی استفاده می کند که آسیب پذیری های امنیتی را کاهش می دهد.*

اول، نرم افزار تا آنجا که ممکن است به اجزای نرم افزار بسیار قابل اعتماد و/یا منبع باز برای احراز هویت، رمزگذاری و ارتباطات متکی است. اینها شامل OpenSSH برای ایجاد اتصالات SSH و تولید جفت کلید خصوصی–عمومی، opensslبرای تولید کلیدهای خصوصی X.509 و درخواست‌های امضای گواهی، GitHubبرای ذخیره کلیدهای عمومی، درخواست‌های امضای گواهینامه و گواهی‌ها وAWS Iot Core ، کارگزار MQTT ایمن با قدرت صنعتی (اما نه منبع باز). این احتمال آسیب‌پذیری‌های امنیتی را که اکنون شناخته شده یا شناخته شده‌اند و به سرعت برطرف نشده‌اند، کاهش می‌دهد.

دوم، اتصال SSH بین یک کامپیوتر کنترل کننده و یک کامپیوتر راه دور تایید و رمزگذاری شده است. داده‌هایی که دیمون SSH و هسته روی پراکسی بین دو تونل انتقال می‌دهند، بخشی از لایه انتقال است که اتصال SSH از آن استفاده می‌کند. این داده ها شکل رمزگذاری شده ترافیک در اتصال SSH هستند، نه متن واضح. سوم، کلیدهای خصوصی SSH و X.509 بر روی رایانه‌هایی که احراز هویت می‌شوند تولید می‌شوند و هرگز از طریق اتصالات شبکه منتقل نمی‌شوند. این احتمال نشت یک کلید مخفی در حین انتقال یا ذخیره نامناسب را کاهش می دهد. به فایل‌های کلید مخفی نام‌هایی با پسوندهایی داده می‌شود که به وضوح آن‌ها را مشخص می‌کند و با پسوندهای کلیدهای عمومی و گواهی‌ها متفاوت است.

چهارم، مدیران و کاربران کلیدهای خصوصی را تنها از طریق دستوراتی با معنای سطح بالا دستکاری می‌کنند که این امکان را از بین می‌برد که کلیدهای مخفی هنگام صدور فرمان اشتباه سطح پایین توسط کاربر، نشت کند. دستورات سطح بالایی برای تولید کلیدهای SSH و X.509 و دستوراتی وجود دارد که از این کلیدها برای ایجاد اتصالات شبکه استفاده می کنند، اما هیچ دستور سطح بالایی وجود ندارد که فایل های کلید مخفی را منتقل یا جابجا کند

پنجم، هر گواهی X.509 در کارگزار MQTT با محدودترین خط مشی دسترسی مرتبط است که همچنان به رایانه اجازه می دهد از این سرویس استفاده کند. به طور خاص، هر رایانه راه دور با دو موضوع منحصر به فرد MQTTمرتبط است، یکی برای دستورات اتصال/قطع اتصال به رایانه راه دور و دیگری برای گزارش وضعیت رایانه راه دور. رایانه های راه دور فقط می توانند در موضوع اول مشترک شوند و فقط می توانند در موضوع دوم منتشر کنند. رایانه های کنترل کننده می توانند در هر یک از موضوعات فرمان در سیستم خود منتشر کنند و فقط می توانند در موضوعات وضعیت همان سیستم مشترک شوند.

ششم، خود نرم افزار منبع باز است و از این رو برای بازرسی و تجزیه و تحلیل باز است و احتمال آسیب پذیری های پنهان را کاهش می دهد. [۸].

## ۴- تضمین های امنیتی سازه

*طراحی نرم افزار چندین تضمین امنیتی مهم را ارائه می دهد.*

اول، دسترسی به پروکسی، حتی دسترسی ریشه، به هیچ یک از رایانه های کنترل کننده یا راه دور اجازه دسترسی نمی دهد، زیرا هیچ کلید خصوصی روی پراکسی ذخیره نمی شود. دسترسی ریشه به پراکسی همچنین محتوایی را از اتصالات بین رایانه های کنترل کننده و راه دور منتشر نمی کند، زیرا شبح SSH و هسته فقط ترافیک رمزگذاری شده را از طریق تونل ها می بینند.

ثانیاً، دسترسی به یک رایانه راه دور، دسترسی به هیچ رایانه کنترلی یا رایانه راه دور دیگری را نمی دهد، زیرا هر رایانه در سیستم یک کلید خصوصی SSH را ذخیره می کند و کلیدهای خصوصی رایانه های راه دور فقط به کاربر عادی دسترسی به پروکسی را می دهد [۹].

سوم، مهاجمی که به یک رایانه راه دور دسترسی پیدا کرده است، نمی تواند دستورات MQTT را برای رایانه های دیگر مشاهده کند و نمی تواند دستورات را منتشر کند. می‌تواند دستورات اتصال/قطع اتصال به آن رایانه را مشاهده کند و می‌تواند پیام‌های وضعیت نادرست، از جمله پیام‌های وضعیت با فرمت نادرست را گزارش کند.

## ۵- هزینه های جاری

هزینه های هر کامپیوتر از راه دور با اجرای این نرم افزار ناچیز است. آمازون حدود ۰,۰۴ دلار (همه مبالغ دلار به دلار آمریکا اشاره دارد) در سال برای اتصال ۷/۲۴ IoT Core هزینه می کند. رایانه های راه دور نیز هر ساعت یک پیام وضعیت ارسال می کنند که به کمتر از ۱۰۰۰۰ پیام در سال می رسد. آمازون ۱ دلار یا کمتر به ازای هر میلیون پیام دریافت می کند، بنابراین به ازای هر رایانه راه دور کمتر از ۰,۰۱ دلار در سال اضافه می کند. این نرم افزار از هیچ خدمات اضافی IoT Core استفاده نمی کند، بنابراین هزینه کل به ازای هر کامپیوتر راه دور در سال حدود ۵ سنت است.

هزینه کرایه یک میزبان پرش در فضای ابری معمولاً قابل توجه تر است، اما همچنان پایین است. ما یک قلمرو را با استفاده از سرور مجازی Amazon Lightsail با ۱ هسته، ۱ گیگابایت حافظه، ۴۰ گیگابایت حافظهSSD ، پهنای باند ۱ ترابایت در ماه و یک آدرس IP ثابت با ۶۰ دلار در سال اجرا می کنیم. برای قلمرویی با مثلاً ۱۰۰ رایانه از راه دور، هزینه هر راه دور بسیار بیشتر از اتصال هسته IoT است، اما برای قلمرویی با ۱۰۰۰ رایانه از راه دور این دو هزینه مشابه هستند.

این هزینه ها همچنین به وضوح نشان می دهد که دلیل مالی برای عدم استفاده از تونل سازی بر اساس تقاضا وجود ندارد. هزینه اضافی ناچیز است.

## ۶- نتیجه گیری

این نرم افزار نسخه بالغ نرم افزاری است که ما به مدت ۹ سال از آن برای دسترسی به ایستگاه های پایه توزیع شده جغرافیایی سیستم های ردیابی حیات وحش ATLAS در چندین کشور و قاره استفاده کرده ایم. ایستگاه های پایه ATLASشامل یک گیرنده رادیویی نمونه (حسگر) و یک کامپیوتر با لینوکس است. ایستگاه های پایه پینگ های رادیویی را از فرستنده های مینیاتوری متصل به حیوانات وحشی دریافت و پردازش می کنند. آنها گزارش های تشخیص را از طریق اتصالات داخلی سلولی به سرور ارسال می کنند تا به سرور اجازه دهند فرستنده و حیوان را بومی سازی کند. اساساً اتصال به رایانه های ایستگاه پایه برای تشخیص و اصلاح خطاها و به روز رسانی نرم افزار ضروری است. سیستم‌های ATLAS با استفاده از نسخه‌های اولیه این نرم‌افزار داده‌هایی را تولید می‌کردند که در چندین نشریه علمی مهم از جمله در Science وPNAS ، استفاده می‌شد. برخی از سیستم‌های ATLAS از مکانیسم‌های اتصال از راه دور جایگزین استفاده می‌کنند، مانند سایر سیستم‌های سنجش علمی. ما معتقدیم که این جایگزین‌ها نسبت به نرم‌افزار ما پایین‌تر هستند و در اینجا دلیل آن را توضیح می‌دهیم. یکی از جایگزین‌هایی که در یکی از سیستم‌های ATLAS که در آن ایستگاه‌های پایه ویندوز را اجرا می‌کردند، استفاده شد، یک راه‌حل تجاری

از راه دور به نام TeamViewer بود. این کار کرد اما به دلیل این واقعیت که اتصال دارای یک رابط کاربری گرافیکی است، نه یک رابط دستوری متنی (یک پوسته) بسیار کند بود. جایگزین دیگری که توسط سیستم ATLASدیگری استفاده می شود، شبکه خصوصی مجازی اختصاصی (VPN) است که از OpenVPN استفاده می کند. نقطه ضعف اصلی این رویکرد نیاز به مدیریت یک مرجع گواهی (CA) است که انجام ایمن و قابل اعتماد دشوار است. همچنین، یک VPN نسبت به راه حل ما امنیت کمتری دارد، زیرا تمام پورت های ایستگاه پایه در معرض مهاجمی هستند که کنترل ایستگاه پایه دیگری را در همان VPN به دست آورده است. این مورد در راه حل ما نیست که در آن به دست آوردن کنترل یک ایستگاه پایه یا پراکسی فقط پورت SSH سایر ایستگاه های پایه را آشکار می کند. این پورت معمولاً به خوبی سخت شده است. جایگزین‌های تجاری نیز وجود دارد که اساساً همان عملکرد نرم‌افزار ما را ارائه می‌کنند، مانند AWS IoT Secure Tunneling، اما به نظر نمی‌رسد استفاده از آن آسان‌تر باشد و بسیار گران است، به ازای هر اتصال به یک کامپیوتر از راه دور ۱ دلار (اتصال را می‌توان باز کرد؛ و تا ۱۲ ساعت بسته است).

## منابع

[1] Yan Y. Wu Z. Wu X. Zhou X. Weng C. A Linux-based integrated structural health monitoring system for bridges in remote regions Instrum. Mesures, Métrol. 18 (6) (2019), pp. 527-534,
[2] A. Weller, Y. Orchan, R. Nathan, M.C.A.J. Weiss, S. Toledo, Characterizing the Accuracy of a Self-Synchronized Reverse-GPS Wildlife Localization System, in: Proceeings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN, Vienna, Austria, 2016,
[3] T. Ylonen, SSH - Secure Login Connections Over the Internet, in: Proceedings of the 6th USENIX Security Symposium, San Jose, CA, 1996, URL.
[4] Dusi M. Gringoli F. Salgarelli L. A preliminary look at the privacy of SSH tunnels Proceedings of 17th International Conference on Computer Communications and Networks (2008),
[5] Lourie E. Schiffner I. Toledo S. Nathan R. Memory and conformity, but not competition, explain spatial partitioning between two neighboring fruit bat colonies Front. Ecol. Evol. 9 (732514) (2021),
[6] Vilk O. Orchan Y. Charter M. Ganot N. Toledo S. Nathan R. Assaf M.
Ergodicity breaking in area-restricted search of avian predators Phys. Rev. X, 12 (2022),
[7] Roeleke M. Schlägel U.E. Gallagher C. Pufelski J. Blohm T. Nathan R. Toledo S. Jeltsch F. Voigt C.C. Insectivorous bats form mobile sensory networks to optimize prey localization: The case of the common noctule bat Proc. Natl. Acad. Sci. (PNAS), 119 (33) (2022),
[8] A. Corl, Charter M. Rozman G. Toledo S. Turjeman S. Kamath P.L. Getz W.M. Nathan R. Bowie R.C.K. Movement ecology and sex are linked to barn owl microbial community composition Mol. Ecol. 20 (7) (2020),
[9] S. Toledo, Shohami D. Schiffner I. Lourie E. Orchan Y. Bartan Y. Nathan R. Cognitive map-based navigation in wild bats revealed by a new high-throughput tracking system Science, 369 (6500) (2020).

# SSH tunnel to connect to remote computers

Mohsen Saeedzadeh *

**abstract**

The purpose of this research is the SSH tunnel to connect to remote computers, Secure Shell (SSH) provides a secure method for remote access to servers, mainly used in Unix and Linux environments. It enables encrypted command line access, which allows secure file transfer, remote command execution, and tunneling.We provide software for establishing secure shell (command line) connections to remote computers for maintenance We provide configuration or debugging, even when remote computers are connected to the Internet so that incoming connections are not allowed. The software is easy to install, has few dependencies, offers very useful security guarantees, and scales to large numbers of remote computers, possibly split into isolated groups. This software relies on one or two cheap cloud services. An early version of the system has been used for nine years to access remote geospatial computers that act as sensors in wildlife tracking systems.

**Keywords:** remote access, secure shell, SSH tunnels,