امیرحسین شریف ۹۴۱۴۳۰۱۶ تبادل کلید دیفی-هلمن

مقدمه

پروتکل تبادل کلید یا پروتکل دیفی – هلمن که ، برای این که دو نفر یا دو کاربر در یک شرکت یا سازمان یا یک شبکه بدون آشنایی قبلی یک کلید رمز مشترک را ایجاد و آن را از طریق یک مسیر ناامن بین خود تبادل نمایند. این پروتکل،اولین روش علمی است که مطرح شده برای تبادل کلید رمز از طریق یک مسیر یا کانال ناامن.

این پروتکل در سال ۱۹۷۶ توسط توسط دو دانشمندرمزشناس به نام های ویتفلید دیفی و مارتین هلمن طراحی و در قالب یک مقاله علمی منتشر شد.

با انتشار این پروتکل پایه اولیه ی رمز های نامتقارن شکل گرفت و بعدا فعالیت های رالف مرکل تکمیل شد.

با گسترش تدریجی رمز های نامتقارن پروتکل های زیادی با استفاده از پروتکل دیفی – هلمن و با قابلیت های بیش تر نسبت به طراحی نیز ساخته شد.

الگوریتم تبادل کلید دیفی- هلمن

مراحل این الگوریتم را با استفاده از یک مثال بیان می شود. فرض کنید که دو کاربر Alice و Bob داریم که می خواهند به یک کلید مشترک به توافق برسند یا می خواهند کلید رمز را بین خودشان با استفاده از یک کانال ناامن به هم دیگر برسانند.

در مرحله ی اول الگوریتم دیفی – هلمن یک عدد اول p را به صورت دلخواه و رندوم انتخاب می کنیم. فرض کنید ، در مثال p=۳۷ در نظر می گیریم و بعد یک عدد x را که به صورت رندوم از بازه ی [۲,p-۲] انتخاب می کنیم. در مثال x = ۵ در نظر گرفته و این دو مقدار را به صورت عمومی است و آن را نیز اعلام می کنیم.

از این جا به سمت کاربر Alice رفته که ببینیم چه محاسباتی را انجام می دهد.

در مرحله دوم، Alice مقدار a رندوم از بازه ی [۲,p-۲]، انتخاب کرده.

در مرحله سوم از این فرمول کاربر Alice مقدار Aرا حساب کرده و به طرف کاربر Bob می فرستد.

$A = x^a \pmod p$

در مثال فرض کنید a=۴قراردادیم پس

$a=۴ \Rightarrow A = ۵^۴ \bmod ۳۷ = ۳۳$

حال در طرف Bob نیز دقیقا مشابه کاربر Alice می باشد.

مرحله چهارم و پنجم ، مقدار b رندوم از بازه ی [۲,p-۲]، انتخاب کرده و سپس با استفاده از فرمول Bرا محاسبه کرده و آن را برای کاربر Alice می فرستیم.

$B = x^b \pmod p$

$b=۸ \Rightarrow B=۵^۸ \bmod ۳۷ =۱۶$

مرحله ششم-حال مقادیر A,Bبرای دوکاربر فرستاده شده است ، در این مثال مقدار Bبرای Alice و مقدار A برای کاربر Bob فرستاده شده است.

مرحله هفتم- حال با دو فرمول زیر مقادیر z,z` را بدست آورده که باهم برابرند و این همان راز یا کلید مشترک است. مقدار z برای کاربر Aliceو z`را برای کاربر Bob در نظر می گیریم. اگر z,z` باهم برابرنباشند یعنی در محاسبات اشتباهی رخ داده است باید دوباره چک شود.

$z = B^a \pmod p \Rightarrow z = ۱۶^۴ \bmod ۳۷ = ۹$

$z` = A^b \pmod p \Rightarrow z` = ۳۳^۸ \bmod ۳۷ = ۹$

راز مشترک برابر با ۹ شد. و این کل کاری که در این الگوریتم برای تبادل کلید انجام می شود.

نکاتی در مورد الگوریتم تبادل کلید دیفی – هلمن

۱- شرط صحت الگوریتم تبادل کلید دیفی – هلمن را می توانیم به صورت ریاضی نشان دهیم که به شرح زیر است

$z = B^a \pmod p$, $B = x^b \pmod p$ => $z = (x^b)^a \pmod p$ => $z = x^{ba} \pmod p$

$z` = A^b \pmod p$ , $A = x^a \pmod p$ => $z` = (x^a)^b \pmod p$ => $z` = x^{ab} \pmod p$

$*x^{ab} = x^{ba}$ => $z=z`$

۲- عدد p را که عدد اولی است که مثال بسیار ساده و کوچک در نظر گرفته شده است، اما در پروتکل هایی که از این الگوریتم برای تبادل کلید استفاده می شود،معمولا از مولد های شبه تصادفی برای تولید عدد تصادفی و اول p که به صورت ایمن باشد،عدد p انتخاب می شود، چون اگر به صورت ایمن عدد p انتخاب نشود،ممکن است که امنیت الگوریتم دیفی – هلمن به مخاطره بی افتد و این که برای دو مقدار b وaمهم است زیرا امنیت به این اعدادنیز وابسته است،پس این دو عدد نیز با استفاده از مولد شبه تصادفی انتخاب شوند.

۳- امنیت الگوریتم تبادل کلید دیفی – هلمن وابسته به امنیت لگاریتم گسسته می باشد. در واقع اگر فرض شود در حین تبادل کلید از کانال ناامن یک نفوذگر باشد،آن نفوذگر مقادیر A,B,P,x را دراختیار داشته و برای بدست آوردن a یا b کوچک می بایست یک لگاریتم را محاسبه و بعد mod p بگیرد،که این مسئله بسیار سختی است،نفوذگر نمی تواند آن را حل کند. در حال حاضر،براساس قدرت محاسباتی کامپیوتر امروزی فقط تا ۳۰۰ رقم p و ۱۰۰ رقم a,b می تواند برای شکستن امنیت این پروتکل بکار رود که در عمل یافتن کلید مشترک غیرممکن می سازد.

۴- برای اعداد بزرگ که در این پروتکل به توان اعداد بزرگی می رسند برای محاسبات برروی کاغذ که گاهی اوقات حتی ماشین حساب ها نمی توانند آن را حساب کنند،از الگوریتم square and multiply استفاده می شود.

الگوریتم مجذور و ضرب یا square and multiply

با یک مثال الگوریتم square and multiply را توضیح می دهیم.

فرض کنید می خواهیم $۱۱^{۱۳۹۹} \bmod ۱۳$ را حساب کنیم،ابتدا عدد ۱۳۹۹ را به مبنا ۲برده و سپس مقداری مانند V را برابر پایه که در اینجا ۱۱ می باشد.در نظر می گیریم.حال که مقدار توان را به مبنا ۲ برده ایم،حال سمت چپ ترین بیت را کنار گذاشته و یکی بعد از آن شروع می کنیم.به ازای هر بیت V به توان ۲ رسانده و mod ۱۳ یا mod عددی که باید را می گیریم.اگر مقدار بیتی که در آن داریم محاسبات را انجام می دهیم،۱ بود،آن گاه V را که به توان ۲ در قبل رسانده بودیم ضرب در پایه که در اینجا ۱۱ می باشد،کرده و mod عددی که باید را می گیریم این مراحل تا به آخرین بیت که همان سمت راست ترین بیت است برسیم و جواب بدست آمده است.حال مثال را ببینید.

$۱۳۹۹ =(۱۰۱۰۱۱۱۰۱۱۱)_۲$

$V = ۱۱ \Rightarrow ۱۱^{۲} \bmod ۱۳ = ۴ \Rightarrow ۴^{۲} * ۱۱ \bmod ۱۳ = ۷ \Rightarrow ۷^{۲} \bmod ۱۳ = ۱۰ \Rightarrow ۱۰^{۲} * ۱۱ \bmod ۱۳ = ۸$
$\Rightarrow ۸^{۲} * ۱۱ \bmod ۱۳ = ۲ \Rightarrow ۲^{۲} * ۱۱ \bmod ۱۳ = ۵ \Rightarrow ۵^{۲} \bmod ۱۳ = ۱۲ \Rightarrow ۱۲^{۲} * ۱۱ \bmod ۱۳ = ۱۱$
$\Rightarrow ۱۱^{۲} * ۱۱ \bmod ۱۳ = ۵ \Rightarrow ۵^{۲} * ۱۱ \bmod ۱۳ = ۲ \Rightarrow ۱۱^{۱۳۹۹} \bmod ۱۳ = ۲$

در هر الگوریتمی که به توان اعداد بزرگ و عملیات mod گیری نیز دارد، می توان از این الگوریتم استفاده کرد.

نحوه پیاده سازی الگوریتم تبادل کلید دیفی – هلمن

در این نرم افزار به جای مقدار x که در اینجا گفته شده از مقدار g استفاده می شود. مقادیر gوp را به صورت دستی وارد کرده و در store variable ذخیره می شود و بعد loadامی شود برای هر دو کاربر Aliceو Bob و دقیقا همان الگوریتم گفته پیاده سازی شده فقط مقادیر رندوم که در دو طرف که هر دو کاربر برای خودشان انتخاب می کردند نیز باید به صورت دستی به آن ها وارد کرد.چند متغییر فقط حروف شان تغییر کرده ، و مقادیر را با یک دیگر جابه جا کرده هردو کاربر و در نهایت با استفاده از چند number operation عملیات به توان رسانی و modگیری و پیدا کردن راز مشترک را انجام می شود و فقط با استفاده یک مقایسه گر یا comparator نشان می دهد که دو مقدار مساوی است یا خیر که بااستفاده یک خروجی Boolean نشان داده می شود و هم چنین با استفاده از خروجی های اخر می توان کلید مشترکی در دو کاربر محاسبه می کنند به صورت جداگانه را می توان مشاهده کرد.

نتیجه گیری

از الگوریتم تبادل کلید دیفی – هلمن استفاده های زیادی می شود،مخصوصا در شبکه یا سازمان های بزرگ و به دلیل نامتقارن بود موفق بوده و تا الان با استفاده از آن توانستند الگوریتم های توسعه یافته یا پیشرفته آن را بسازند و استفاده کنند،پس به نظر می رسد که در سال های آتی نیز این الگوریتم ممکن است پیشرفته نیز بشود ولی در حال حاضر نتوانستند این الگوریتم در حملاتی به طور مستقیم مورد شکسته شدن قرار بگیرد.

منابع

Wikipedia

ویدیو های آموزشی درس رایانش امن