2024

Secure Software Development

END TO END ENCRYPTION

NAREK OGANES ERVANDIAN

Indhold

Problemformulering

Er det muligt at kommunikere med en tilfældig person over tekstbeskeder på et offentligt netværk uden at nogen tredjeparter kan få kendskab til indholdet af samtalen? Denne problemformulering undersøger, hvilke teknologier og metoder der kan anvendes for at sikre fuldstændig fortrolighed i tekstbaseret kommunikation mellem to personer, som ikke kender hinanden på forhånd og ikke har mødt hinanden fysisk, på en måde hvor ingen udenforstående kan overvåge eller afkode beskederne.

Introduktion

Kommunikation over et offentligt netværk kan blive opfanget og læst af uvedkommende, hvilket gør det usikkert at sende private beskeder over et netværk. For at løse denne problemstilling, kan der bruges kryptografering til at kryptere beskeder inden de sendes afsted, dog består udfordringen her i hvordan to personer kan kryptere og dekryptere beskeder uden at mødes med hinanden fysisk for at aftale en bestemt måde at kryptere og dekryptere på og uden at sende krypterings nøglen over et usikkert netværk. Denne problemstilling har Difffie Hellman Key exchange en løsning på.

Diffie Hellman Key exchange

Diffie-Hellman Key exchange er en metode til sikker udveksling af kryptografiske nøgler over en offentlig kanal. Det giver to personer mulighed for at etablere en fælles hemmelig nøgle, som kan bruges til krypteret kommunikation, uden at det er nødvendigt at transmittere nøglen direkte. Dette opnås ved at begge parter eller personer får en public nøgle og en privat nøgle. Disse bruges til at beregne den delte hemmelige nøgle, som bruges til kryptering og dekryptering.

Dannelsen af den delte hemmelige nøgle fungere på følgende måde:

Begge parter bliver enige om et stort primtal p og et grundtal g også kaldt generatoren, hvor g er en primitiv rod modulo. Disse værdier behøver ikke at være hemmelige og kan deles åbent. Hver person genererer en privat nøgle. Person A genererer en privat nøgle a, som er et tilfældigt stort tal og person B genererer en privat nøgle b, som også er et tilfældigt stort tal. Hver part beregner deres public nøgle ved hjælp af de aftalte værdier p og g. A og B udveksler nu deres offentlige nøgler over den offentlige kanal hvorefter hver part bruger deres egen private nøgle og den anden parts offentlige nøgle til at beregne den delte hemmelige nøgle. På grund af modulær aritmetik resulterer begge beregninger i den samme fælles hemmelige nøgle som kan bruges som en symmetrisk nøgle til yderligere krypteret kommunikation.

Man in the Middle udfordring

Diffie Hellman alene er ikke sikkert nok i sig selv da det er sårbart over for Man in the Middle angreb og ikke designet til at beskytte imod det. Man in the Middle angreb sker når en person sidder mellem person A og B og lytter på trafikken. Angriberen kan udgive sig for at være A eller B og sende beskeder som var angriberen enten A eller B. Et scenarie for dette er følgende:

Person A og B udveksler public nøgler, som angriberen opfanger. Angriberen sender nu dens egen public nøgle til A og B og udgiver sig for at være A over for B og B over for A. Nu opretter angriberen en delt hemmelig nøgle med A og en delt hemmelig nøgle med B, da der nu sker en Diffie Hellman key exchange. Nu har angriberen en delt hemmelig nøgle med både A og B. Når A sender en besked, opfanger angriberen

det, dekryptere det og kryptere det igen med den delte hemmelige nøgle af B og sender beskeden. Her kan angriberen rette beskeden inden den bliver sendt afsted.

Der er altså ingen authenticity verifikationer i Diffie Hellman key exchange. For at beskytte imod Man in the Middle angreb bruges der typisk teknologier som RSA.

RSA imod Man in the Middle

RSA (Rivest-Shamir-Adleman) bruger digital signatur og PKI (Public Key Infrastructure) med certifikater til at bekræfte autenticiteten mellem afsender og modtager. Ideen med en digital signatur er, at bruge et RSA nøglesæt af en public nøgle og privat nøgle, til at bekræfte autenticitet. Det bruges på følgende måde:

Afsenderen krypterer sin besked med modtagerens public nøgle. Den krypterede besked kan kun dekrypteres af modtageren ved hjælp af modtagerens private nøgle. Herefter signerer afsenderen beskeden ved at generere en hash af beskeden og kryptere hash'en med sin egen private nøgle. Denne signatur kan kun verificeres med afsenderens public nøgle. Den krypterede besked og den digitale signatur sendes til modtageren.

Modtageren dekrypterer beskeden med sin private nøgle. Modtageren dekrypterer signaturen med afsenderens public nøgle for at få hash'en af beskeden. Modtageren genererer nu en hash af den dekrypterede besked og sammenligner den med den dekrypterede hash fra signaturen. Hvis de matcher, er beskeden uændret og autentisk.

Dette sikrer, at beskeden virkelig stammer fra den person, der sendte den, da kun denne person har den private nøgle, der matcher den public nøgle i certifikatet. Det sikrer også, at beskeden ikke er blevet ændret undervejs, da hash-værdierne matcher.

PKI (Public Key Infrastructure) er et rammeværk, der bruger digitale certifikater til at sikre kommunikation og bekræfte identiteter i et netværk. Certifikaterne udstedes af en betroet tredjepart kaldet en certifikatmyndighed (CA).

Når en bruger ønsker at sikre sin kommunikation eller bekræfte sin identitet, anmoder de om et certifikat fra CA'en. Certifikatet indeholder brugerens public nøgle og identitetsoplysninger, samt CA'ens digitale signatur for at bekræfte certifikatets ægthed.

Diffie Hellman og RSA kombineret

Det er sjældent af Diffie Hellman bliver brugt alene, da der som nævnt er udfordringer med Man in the Middle angreb, men RSA er heller ikke perfekt uden Diffie Hellman. Hvis RSA bliver brugt uden Diffie Hellman, opstår der en single point of failure, som er serveren. Hvis angriberen får adgang til de private nøgler, har angriberen pludselig adgang til at læse alle samtaler der har været mellem de to parter. Endvidere har RSA-nøgler typisk meget længere gyldighedsperioder, mens Diffie-Hellman er baseret på en sessionsbaseret tilgang. De delte hemmelige nøgler fra Diffie Hellman bør ikke gemmes i nogen databaser og bør kun bruges i de enkelte sessioner.

Implementering

Kode implementeringen er fortaget i en Blazor Server applikation skrevet i C# .net 8. Applikationens hovedformål er at kunne tillade brugere at sende hinanden private beskeder som automatisk krypteres og dekrypteres ved hjælp af Diffie Hellman mønsteret.

ChatHub.cs – Signal R

Applikationen burger Signal R, som er et Microsoft bibliotek, til at sende realtids beskeder ud til web klienter. I Chathub'en, som er en C# klasse som implementere Hub interfacet, defineres en række funktioner, som klienter subscribes til, som skal pushes ud til dem. Et eksempel her er at når der er en bruger der kommer online, så bliver alle andre klienter automatisk informeret med beskeden. På klient siden er det så muligt at tilføje logik omkring, hvad der præcist skal ske når Signal R kommer med et signal. I Chathub.cs klassen er der logik til at håndtere at brugere logger ind og ud, at sende og requeste public nøgler og sende private beskeder.

Index.razor – MainPage

Når brugeren kommer ind på hoved siden, bliver brugeren præsenteret for egen brugernavn og alle andre brugere der er connected og er online. Nu skal brugeren så vælge, en af dem som er online, som brugeren gerne vil sende en privat besked til. Brugeren klikker på en anden online bruger og kan nu sende en privat besked. I det øjeblik brugern vælger en anden bruger, sker der en Diffie hellman nøgle udveksling mellem de to brugere. Her bliver de to brugeres public nøgler samt IV'er udvekslet ved hjælp af Signal R og vil nu automatisk kryptere og dekryptere beskederne når de sendes. Kryptering og dekryptering sker på klient siden og sker ved hælp af en instance af klassen DiffieHellmanKeyExchange.cs.

DiffieHellmanKeyExchange.cs – Kryptering

Klassen DiffieHellmanKeyExchange står for at oprette private og public nøgler, kryptere og dekryptere beskeder. Så snart en ny instance af denne klasse oprettes, oprettes automatisk et sæt af privat og public nøgle, da denne logik er sat i konstruktøren. Her bruges ECDiffieHellmanCng objektet til at generere nøglerne med. KeyDerivationFunction sættes til ECDiffieHellmanKeyDerivationFunction.Hash, og hash-algoritmen sættes til SHA-256. Hver gang siden chat siden initialiseres oprettes en ny instance af klassen, i denne udgave af programmet.

Ved at have en variable til DiffieHellmanKeyExchange kan public nøglen, IV, krypteringsfunktionen og dekrypteringsfunktionen tilgås. Hver gang der skal krypteres for at sende en besked afsted, oprettes en ny IV og sendes sammen med beskeden. Når der krypteres bruges metoden AES til at kryptere som anvender en offentlig nøgle af modtageren og den private nøgle af afsenderen. Det bliver konverteret til en CngKey. Den delte hemmelige nøgle bliver oprettet ved at diffieHellman.DeriveKeyMaterial bruger public nøglen af modtageren som bliver brugt som AES nøgle. Dekryptering bruger IV af krypteringen samt public nøglen og bruger AES til at dekryptere.

I traditionel Diffie-Hellman nøgleudveksling bruges parametrene g (generatoren) og p (primtallet) til at generere private og offentlige nøgler. Men i ECDiffieHellmanCng biblioteket bruges en variant af Diffie-Hellman som kaldes Elliptic Curve Diffie-Hellman (ECDH). ECDH bruger elliptiske kurver i stedet for traditionelle store primtal, og derfor er g og p ikke nødvendige i kode implementeringen.

konklusion

Det muligt at sende private beskeder til en tilfældig person uden at nogen tredjeparter kan opfange og aflæse beskederne ved hjælp af Diffie Hellman nøgle udvekslings mønstret. Ved hjælp af Diffie Hellman behøves der heller ikke aftales en krypteringsmetode på forhånd ved at mødes fysisk eller over et netværk, da det ved brug af private og public nøglerne af afsender og modtager kan kryptere og dekryptere beskeder sikkert.