Tidlig Rapport – Semesteroppgave i datasikkerhet

Harald Dahle (795955) og Joakim L. Gilje (796196)

Sammendrag
Oppgaven går ut på å implementere RSA-krypteringen. Deloppgaver for denne krypteringen er å implementere nøkkelgenerator og selve krypteringen. Det å implementere en nøkkelgenerator for RSA-krypteringen krever igjen at det er tilgjengelig en funksjon for å finne primtall.

Da primtall er svært sentralt for offentlig-nøkkel kryptering generelt, har vi valgt å implementere en primtallstester i oppgaven.

Da offentlig-nøkkel kryptering involverer bruk av store tall (som går langt utover 32 og 64-bits), har det også vært bruk for funksjonalitet for å jobbe med så store tall. Vi anså det som langt utenfor rammene av oppgaven vi har godkjent å implementere et slikt bibliotek. På grunn av dette valgte vi å se etter et bibliotek og funksjoner for å jobbe mot svært store tall fra en tredje part.

Hva har vi implementert til nå
Vi bestemte oss tidlig for å implementere oppgaven i C, uten at vi har noen spesiell grunn til dette annet enn at vi har tidligere erfaringer med språket.

Til nå har vi konsentrert oss om å implementere de sentrale delene av RSA-krypteringen. Med dette mener vi primtallstesteren og «modular exponentiation» ($a^{(b)} \bmod c$). Primtallstesteren er naturligvis en sentral del i nøkkelgeneratoren, mens $a^{(b)} \bmod c$ jo er selve kryptering og dekrypteringen i RSA. Ved innleveringen av denne rapporten er disse to funksjonene ferdig implementert for 32-bits tall.

Deretter har vi konsentrert oss om å finne et bibliotek for å behandle store tall. Valget falt på OpenSSL. Dette biblioteket har den fordelen at det er inkludert som standard i de aller fleste UNIX og UNIX-lignende systemer.

OpenSSL i seg selv inkluderer alle funksjoner for RSA-kryptering, inkludert nøkkelgenerator og primtallstester. Av koden vil det gå frem at vi kun benytter oss av funksjoner relatert til store tall. Spesifikt kalles denne delen for BN i OpenSSL.

Veien videre
Vi må fullføre jobben med å konvertere koden vår til å benytte BN-funksjonene til OpenSSL. Deretter kan vi fullføre nøkkelgeneratoren. Til slutt må vi implementere funksjonalitet for å jobbe på blokker av tekst for deretter å sende blokkene til enten kryptering eller dekryptering.

Koden vår ligger under CVS-kontroll på http://jgilje.net/cgi-bin/cvsweb/RSAcrypt/.