Open-Source Koddosa

Ett säkrare login på Linux-system

Johan Ben Mohammad
Adam Fredriksson
Christoffer Mathiesen
Eliot Roxbergh
Gustav Örtenberg

24 mars 2017

Projektroll	Efternamn
Sammankallande	Mathiesen
Versionshanteringsansvarig	Roxbergh
Loggbokshanterare	Fredriksson
-	Ben Mohammad
-	Örtenberg

1 Bakgrund

Detta projekt ämnar att förbättra säkerheten vid inloggning på datorer som innehåller värdefull information. Den tänkta målbilden är Linux-system som kommer att användas av forskningspersonal på Chalmers tekniska högskola. Om datan på de systemen skulle kommas åt av utomstående kan hela projekt bli stulna eller värdelösa, därför är det intressant att ha en extra nivå av säkerhet för att logga in på datorerna.

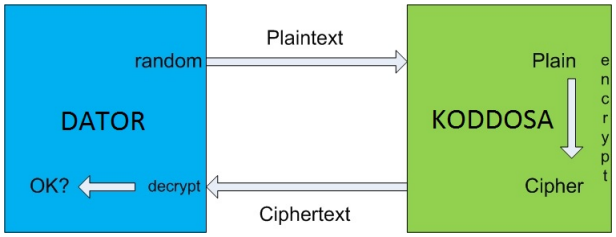
Tanken är att använda en koddosa som extra autentisering vid inloggning. Detta kallas tvåfaktorsautentisering, en metod som banker har använt i flera år för att öka säkerheten på dess tjänster. I detta projektet är den ena faktorn är ett lösenord och den andra är dosans challenge-response-funktion. Datorn genererar en kod som anges på dosan och i respons fås en text som man skriver in i datorn, denna text är krypterad med dosans unika nyckel. Koddosan ökar säkerheten då det krävs både lösenord och dosa, dvs en hemlighet och en fysisk entitet. Även om en angripare har full översikt över datorn och dess knapptryckningar så är detta inte tillräckligt då även dosan krävs för att logga in. Eftersom man ej vet om dosans krypteringsnyckel kan man ej förutsäga vad svarstexten kommer vara.

2 Syfte

Syftet med projektet är att skapa en unik koddosa som ger ett extra steg vid inloggning till ett Linux-system. Med en dosa ökas säkerheten då det krävs både dosan och lösenord för att logga in. Tillsammans med en egenutvecklad autentiseringsmodul för Linux görs en tvåfaktorsautentisering för att kunna logga in i systemet. Med detta prototypsystem som bas kan det skapas fler dosor för flera användare, t.ex. på företagsnivå.

3 Problemuppställning

Projektet består av flera olika arbetsområden som i slutet sätts ihop till en komplett tvåfaktorsautentiseringsenhet.



Figur 1: Flödesbild för kommunikation mellan PAM och FPGA

Figur 1 visar översiktligt hur datorn ska kommunicera med dosan. Datorn genererar en slumpmässig text som skrivs in på dosan. Texten krypteras och visas på dosans skärm. Den krypterade texten skrivs in på datorn där den dekrypteras och används för att autentisera användaren. Om den dekrypterade texten matchar originaltexten tillåts användaren att logga in med sina uppgifter.

3.1 Kryptering

Det är viktigt att väjla en krypteringsalgoritm som är asymmetrisk, d.v.s. den ena nyckeln kan vara hemlig medan den andra är känd. Detta för att endast personen med den hemliga nyckeln kan skicka ett krypterat meddelande som kan dekrypteras med den kända nyckeln och således verifiera att det var ägaren av den hemliga nyckeln som skickade meddelandet. Olikt symmetriska krypton där samma nyckel används för både kryptering och dekryptering. Krypteringsalgoritmen bör även uppfylla kravet att det ska vara extremt tids- och prestandakrävande att lista ut den hemliga nyckeln.

Utifrån dess krav valdes RSA(Rivest Shamir Aldeman)-algoritmen för kryptering.[2] RSA bygger på att två stora primtal väljs ut och deras produkt används för att genera ett asymmetriskt nyckelpar. Säkerheten ligger i att det är svårt att primtalsfaktorisera stora tal.

En avvägning som skall göras är längden på krypteringsnycklarna, dvs storleken på primtalen. En längre nyckel är svårare att knäcka. Däremot kräver längre nycklar mer prestanda i hårdvaran vid inloggning. Saker att tänka på vid denna avvägning är användarvänlighet, säkerhet samt prestandakostnad.

3.2 Kommunikationen med PAM

Lösenordsautentiseringen kommer ske med hjälp av en egenutvecklad PAM(Pluggable Authentication Module). Med PAM kan man skapa olika services för bland an-

nat autentisering, lösenordslagring samt användarbehörighet.[4]

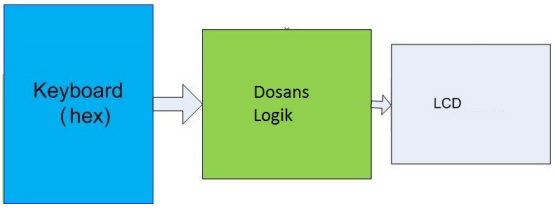
PAM gör det möjligt att bygga egna verifieringsmoduler med tillhörande logik. I detta projekt generering av slumptext samt efter dekryptering och verifiering av chiffertext från dosan. Om den dekrypterade chiffertexten ej stämmer överens med slumptexten släpps användaren inte in. För dekryptering krävs att PAM håller i den publika nyckeln i nyckelparet.

När rätt lösenord är angivet visar PAM ett slumptal i hexadecimal form som användaren skriver in på koddosan. Svaret, chiffertexten, från dosan skrivs in på datorn. PAM verifierar sedan chiffertexten med hjälp av den publika nyckeln. Om användarens lösenord stämmer samt verifieringen är lyckad tillåter PAM användaren inloggning på datorn.

3.3 VHDL

Koddosan kommer att programmeras i VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language). Det innebär att goda kunskaper om språket kommer att behövas från åtminstone några utav medlemmarna i arbetsgruppen. VHDL används för att programmera den FPGA(Field Programmable Gate Array) som kommer utgöra koddosans logikkrets.

3.4 Hårdvara



Figur 2: Helhetsbild över hårdvarustrukturen

Initialt ska en prototyp skapas på utvecklingskortet Nexys3 som på vilken FPGA:n sitter.[1] Om det skulle behövas kommer ett annat kort användas. Utvecklingskortet står för beräkningsenheten och minne. Utöver detta behövs ett tangentbord för indata och en LCD-skärm för utdata från kortet.

3.5 Optimering

Några optimeringar som kan tänkas göras med FPGA:n är att öka beräkningseffektivitet, minska strömförbrukningen och minska den fysiska plats som kretsen tar. Dessa tre optimeringar går inte alltid hand i hand och därför kommer kompromisser att behöva göras. Ett mål är att produkten skall vara användarvänlig

på ett sätt som ökar säkerheten men ej tar för lång tid att hantera. Optimeringar görs med hjälp av Xilinx ISE.[5]

4 Avgränsningar

Arbetsgruppen kommer att arbeta med detta projekt från mitten av januari 2017 till och med maj månad. Då projektet är av okänd komplexitet måste avgränsningar göras för att undvika tidsbrist.

Produkten vid projektets slut kommer inte att vara en portabel lösning, utan endast vara en prototyp. Produkten kommer bestå av flertalet kretskort och moduler kopplade till ett FPGA-utvecklingskort (Nexys 3). Detta kommer att vara otympligt att bära med sig. Och då projektet troligtvis kommer att ta för lång tid för att ge tillfälle till skapa en användarvänlig dosa kommer detta att utebli.

Detta projekt kommer endast använda sig av RSA som kryptering. Andra algoritmer kommer ej att testas.

Detta projekt kommer endast använda sig av Nexus3-utvecklingskortet. Dess hårdvarubegränsningar kommer att forma vad som är möjligt att göra inom projektet. Dem enda externa modulerna till utvecklingskortet kommer att vara en tangentbord och en LCD-skärm. Inga andra moduler tillämpas.

Optimering av enheten är inte huvudfokus i projektet. Men om det hinns med kommer mer tid att ägnas åt detta.

Även om prototypen i princip ska kunna integreras med ett större system kommer prototypen endast att kopplas till en testdator. I detta projekt kommer vi inte att beröra systemnivån av problematiken eller integrera oss med ett system/admin.

Gällande PAM-modulen skrivs endast ett tillägg till befintlig PAM-inloggning.

5 Metod

5.1 Grupproller

Gruppen kommer att dela ut uppgifter allteftersom till olika personer och därav blir de personerna även ansvariga för den uppgiften. Dock är en del roller permanenta då de krävs någon ansvarig för dessa under hela projektets gång. De permanenta rollerna är Versionshanteringsansvarig, Sammankallande samt Loggbokshanterare.

5.2 Verktyg

Det som är gemensamt för gruppen är versionshanteringsprogrammet Git, loggbok på Google Docs och kommunikationsappen Telegram. För att skriva rapporter och diverse dokument använts typsättningssystemet LaTeX. Utöver dessa är det upp till varje gruppmedlem att bestämma själv vad för verktyg de vill använda för programmering, dokumentation och annat, förutsatt att de är kompatibla med programvaror som är gratis eller det finns studentlicenser till.

5.3 Kommunikation

Varje vecka kommer gruppen att ha ett uppföljningsmöte på vad som gjorts och vad som behöver göras. Även problem och lösningar kan behöva diskuteras vid uppkomst. Mötena anordnas av Sammankallande. På varje möte kommer en ordförande, en sekreterare samt en justerare att tillsättas. Vid slutet av varje möte bestäms när nästa möte skall ske.

Kommunikationen i gruppen utanför mötena kommer att främst utföras via kommunikationsappen Telegram samt e-mail. Om det av något skäl behövs omedelbar kommunikation kommer telefonsamtal att användas. Öppen kommunikation (så att alla ser konversationen) mellan gruppmedlemmar är uppmuntrat men inte tvunget.

Ifall det anses behövas kan arbetsmöten schemaläggas under uppföljningsmötena. Dessa möten är obligatoriska att komma på om inte på förhand det meddelats att detta inte går. Dessa möten behöver dock inte nödvändigtvis innefatta hela gruppen samtidigt.

5.4 Arbetsmetod

Arbetsuppgifter utförs individuellt eller i mindre grupper på främst det personerna blev tilldelade under uppföljningsmötena, men är inte förhindrade att hjälpa med andras uppgifter. Det förväntas att alla gruppmedlemmar jobbar efter förmåga så mycket de kan på projektet, men absolut minst att de utför deras arbetsuppgift varje vecka. När och var man arbetar är upp till var och en. Ett undantag är arbetsmöten man är kallad till, vilket är obligatoriska.

Varje gruppmedlem är personligen ansvarig för att det de bidrar med är korrekt och väldokumenterat. Med detta innebär det att programkod är väl kommenterad och fungerande, att commits på Git beskriver de tillägg och/eller ändringar som utförts, att dokument är i korrekt format och rättstavade, o.s.v. För att förhindra arbetsuppehåll p.g.a. oförståelse av andras kod skall det finnas en enkel dokumentation. Alla har eget ansvar för att dokumentera sin egen kod och under uppföljningsmötena förklara hur det man skapat fungerar.

Gruppmedlemmarna skall även rapportera in all sin tidsallokering i en grupploggbok i Google Docs.

5.5 Kvalité

För att försäkra att arbetet som utförs är användbart och korrekt kommer en lista med frågeställningar behöva svaras på. Dessa behöver inte dokumenteras, men ska vara i tankarna under arbetets gång.

Vad? Vad är det som ska göras och varför?
Hur? Hur fungerar implemenationen?
Fungerar det? Är arbetet fungerande? Uppfyller den förväntningarna?
Dokumenterat? Har arbetet blivit dokumenterat? Är dokumentationen förståelig?
Uppladdat? Är arbetet uppladdat på Git? Är commitkommentaren beskrivande över vad som utförts?

5.6 Spelregler

För att försäkra engagemang och effektivitet finns ett fåtal regler. Alla gruppmedlemmar förväntas vara trevliga, att själva hålla sig uppdaterade, vara med på, och i tid till, obligatoriska moment samt utföra sina uppgifter. Misslyckande med dessa regler skall diskuteras på uppföljningsmötena, men har vid enstaka gånger inte ett straff. Vid upprepade misslyckanden kan gruppen bestämma ett straff. Ett straff måste dock vara av vänskaplig karaktär, som till exempel bjudande av fika.

6 Tidsplan

På grund av projektets karaktär kom gruppen fram till att en iterativ utvecklingsprocess likt Scrum[3] skulle tillämpas. En anledning är behovet att kunna testa produkten vilket kräver både prototypen och PC tillsammans.

Produkten har två delar, koddosan och inloggningsprogrammet. I tidsplanen nedan heter koddosan VHDL-kodning och inloggningsprogrammet PAM-programering. Dessa två processer överlappar till stor del då datautbytet kräver standardiserad kommunikation mellan enheterna på grund av designval som kommer att göras för koddosan.

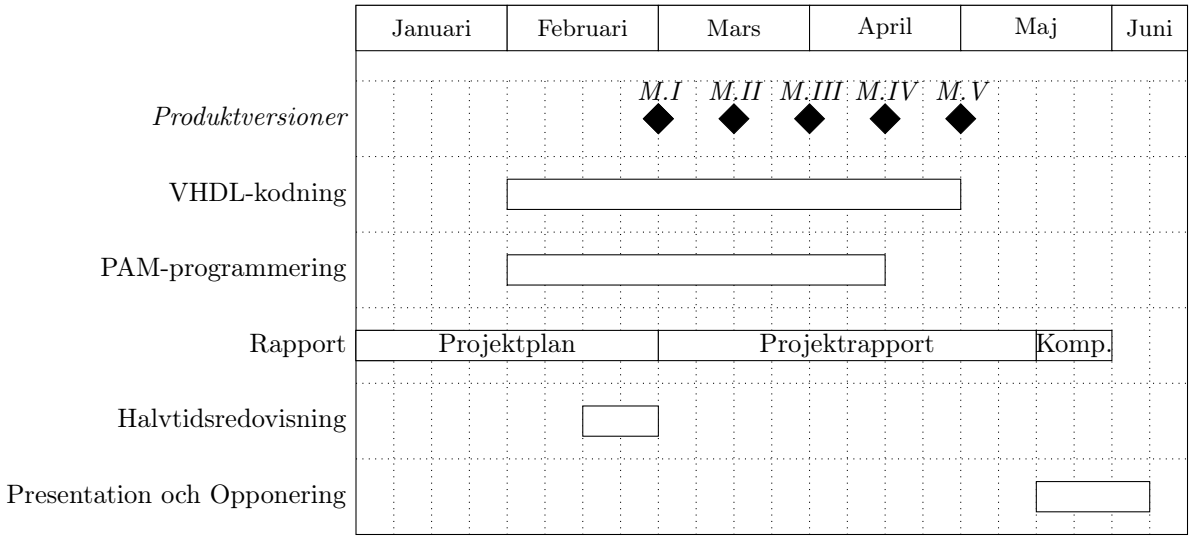
I Man skall kunna mata in siffrorna från testdatorn i koddosan via den knappsats som finns. Från dosan skall man då få på LCD:n de tecken man skall skriva som svar till PC:n. Det finns inget krav på lösenord.

II Implementera RSA kryptografi för både dosan och testdatorn.

III Öka säkerheten från II. Krav på lösenord vid login.

IV Att ha en färdig produkt.

V Fixa eventuella fel med IV.



Figur 3: Övergripande tidsplan

Referenser

[1] Digilent. (2016). Nexys 3^{TM} FPGA Board Reference Manual. Hämtad från: `https://reference.digilentinc.com/_media/nexys:nexys3:nexys3_rm.pdf`. [Online; Läst 2017-02-08].

[2] Borko Furht, editor. *RSA Public-Key Encryption Algorithm*, pages 770–770. Springer US, Boston, MA, 2008.

[3] Ken Schwaber and Jeff Sutherland. (2016). The Scrum Guide. Hämtad från: `http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf`. [Online; Läst 2017-02-16].

[4] TuxRadar. (2009). How PAM works. Hämtad från: `http://www.tuxradar.com/content/how-pam-works`. [Online; Läst 2017-02-10].

[5] [Online; Läst 2017-02-20] Xilinx. (2017). ISE Design Suite. Hämtad från: `https://www.xilinx.com/products/design-tools/ise-design-suite.html`.