TEKNIK FÖR SAMLAD INLOGGNING

Niklas Björk, 33077
Kandidatavhandling i datateknik
Åbo Akademi
Institutionen för Informationsteknologi

Referat

Samlad inloggning (Single Sign-On, SSO) är benämningen för tekniken som eliminerar kravet på att användarna skall ha flera lösenord för att komma åt de olika dataresurserna inom ett eller flera nätverk. Detta innebär att data hjälpcentralen kommer att få mindre arbeten angående lösenords återställning och data administrationen behöver inte investera lika mycket tid på att upprätthålla användar databaser hos de olika data resurserna. I denna avhandling tas upp olika tekniker som möjliggör samlad inloggning och dessa tekniker förklaras mera ingående. Dessutom behandlas olika arkitekturer av samlad inloggning och deras för- och nackdelar jämförs.

Nyckelord: Samlad Inloggning **(**Single Sign-On, SSO), Web samlad inloggning, Legacy/Enterprise samlad inloggning, Gemensam samlad inloggning, Autentisering, Auktorisering

Innehållsförteckning

1. Inledning

Enligt Computer Sweden[1] är den svenska översättningen för Single Sign-On samlad inloggning. Samlad inloggning blev uppfunnen på grund av diverse orsaker. En orsak var att man ville minska på antalet lösenord som en användare måste komma ihåg för att ha tillgång till alla dataresurser i ett nätverk. En användare kan ha upp till ett dussin olika lösenord att komma ihåg och detta ledde till att man började skriva ner dessa lösenord på ”komihåg” lappar. Dessa ”komihåg” lappar sparad man sedan ”lätt åtkomligt” vid sin arbetsstation vilket utgjorde en allvarlig säkerhets risk. En annan orsak som bidrog till att samlad inloggning blev uppfunnen var att man ville lätta på IT-stödets arbetsbörda genom att minska på antalet incidenter som de får för att återställa glömda lösenord. Istället kan de koncentrera sig för verkliga IT problem.

Samlad inloggnings-lösningar är indelade i två huvud grupper, nämligen de som hanterar bara en uppsättning av autentiseringsuppgifter och de som hanterar flera uppsättningar av autentiseringsuppgifter [3]. Dessa lösningar är baserade på redan existerande teknik.

I detta arbete kommer jag att presentera olika arkitekturer av samlad inloggning och jämföra dessas med varandra genom att ta upp fördelar och nackdelar hos dem. Jag tar också upp den alternativa lösningen med identitets synkronisering även om den inte är en sann samlad inloggningslösning.

Jag har använt mig av engelska termer då det inte finns någon klar svensk översättning och dessa termer är skrivna kursivt så att man lättare kan urskilja dessa från texten.

[1]Computer Swedens ordlista är en dataterms ordlista var man har översatt engelska datatermer till svenska.
Källa: http://cstjanster.idg.se/sprakwebben/ord.asp?ord=samlad%20inloggning

2. Samlad inloggning i sin korthet

Samlad inloggning är tekniken som ger användaren möjlighet till en process för engångs autentiserings och sedan ha tillgång till andra skyddade resurser utan att behöva autentisera sig igen. Open Group, som är ett globalt konsortium vars uppgift är att möjliggöra uppnåendet av verksamhetsmål genom IT-standarder, definierar samlad inloggning som den mekanism där en ända handling av användarautentisering och auktorisering kan tillåta användaren att komma åt alla datorer och system var användaren har åtkomstbehörighet, utan att behöva ange flera lösenord [3]. Det bör klargöras att ordet ”användare” skall tolkas i en bred bemärkelse: det täcker alla säkerhetsobjekt som har tillgång till de resurser som är under auktoriserings auktoritets kontroll.

En typisk företagsanvändare måste autentisera sig flera gånger för att kunna få tillgång till diverse applikationer som han/hon behöver för att kunna utföra sitt arbete. Från användarens synvinkel är dessa upprepade autentiseringar och kravet på att komma ihåg flera olika lösenord de ledande orsaker till att de upplever missnöje av att använda företagets datatjänster. Enligt IT-administrationen så ökar incidenter med glömda lösenord markant på administrationens kostnader.

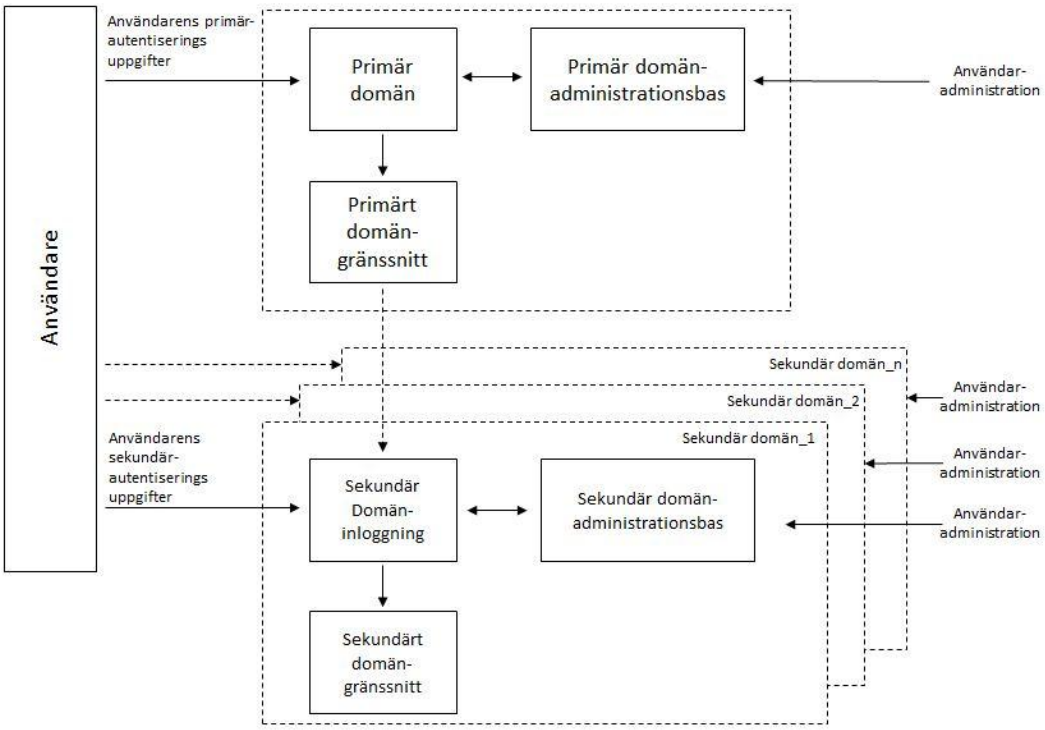
På grund av de till synes oöverkomliga problemen som flera autentiserings identiteter utgör, så har konceptet samlad inloggning blivit den ” heliga graalen” för projekt var man skall hantera olika identitet [6, 8].

2.1 Jämförelse av inloggning utan- och med samlad inloggning

Figur 1 illustrerar ett nätverk som man inte har implementerat en lösning för samlad inloggning och i detta fall så hamnar användaren att upprepade gånger autentisera sig för att nå alla de data resurser som han/hon behöver. Då användaren autentiserar sig för första gången så sker det till en så kallad primär domän, alltså till den domän som är förvald för användaren. För att autentisera sig till denna domän så måste användarnamn och lösenord ges till den primära

domänen som sedan verifierar i sin databas att dessa autentiseringsvärden stämmer för den ifrågavarande användaren. Om autentiseringsvärden var korrekta så får användaren tillgång till alla de dataresurserna som finns till förfogande hos denna domän. Men om användaren är ute efter ett specifikt program eller tjänst som inte finns på denna domän utan på en annan, så hamnar han/hon att ta kontakt med någon annan domän. För att få tillgång till denna dataresurs på den andra domänen så måste användaren autentisera sig igen. Följande domäner som användaren skall autentisera sig till kallas för sekundära domäner.

Figur 1 illustrerar också hur komplext det är att upprätthålla användarkonton för varje enskild domän. Varje domän har sin egen databas som innehåller användaruppgifter och rättigheter på alla de användare som skall ha tillgång till domänen. Om en användare glömt sin lösenord och vill att den skall nollställas så måste IT-stödet manuellt ta kontakt med varje domän och nollställa den specifika användarens lösenord. Samma genomgång gäller om användaren slutar jobba hos företaget, men istället raderas kontot från databasen.

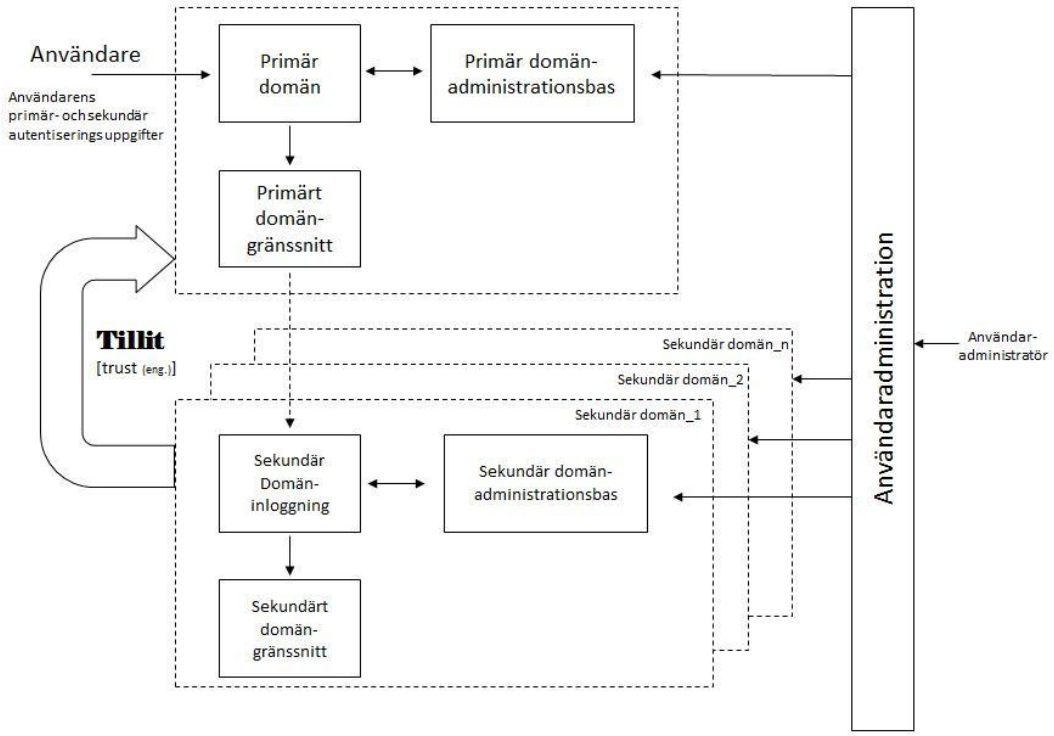


Figur 1[7] Användaren måste autentisera sig till de olika domänerna för att nå all de dataresurser som behövs.

Figur 2 illustrerar ett nätverk var man har implementerat en lösning för samlad inloggning. I detta fall måste användaren autentisera sig endast en gång och detta sker via primärdomänen. Efter en lyckad autentisering har användaren full tillgång till alla domän i nätverket och behöver inte autentisera sig flera gånger. I samma figur ser vi att alla sekundära domäner litar på den primära domänen till vilken användaren autentiserat sig. Primära domänen tar upp användarens autentiseringsuppgifter och kan använda dessa för autentisering till de sekundära domänerna.

Sekundär autentisering kan ske i olika former.
Det kan ske direkt, var informationen som användaren gett överförs till sekundära domänerna i form av sekundär autentisering [7].
Det kan också ske indirekt, var informationen som användaren gett används till att hämta andra användaridentifierings- och autentiseringsuppgifter vilka är sparade i den samlade inloggningsanvändardatabasen. Den hämtade informationen används sedan till sekundär domän autentisering. [7]
Det kan ske omedelbart, för att skapa en session med de sekundära domänerna som en del av den ursprungliga autentiseringssessionen. Detta innebär att klientprogram anropas automatiskt och att kommunikationen fastställs vid samma tidpunkt som den primära autentiseringen [7].
Det kan också ske stegvist dvs. att man sparar i minnet eller mellanlagrar användaridentifiering och autentiseringsvärden som används då en åtkomstbegäran för en sekundär domän gjorts av användaren [7].

En samlad inloggningslösning som denna kommer också att underlätta avsevärt på IT stödets arbete då de har bara ett enda databas att administrera. Denna databas synkroniseras automatiskt till företagets alla andra domän.



Figur2 [7] Användaren autentiserar sig bara till primära domänen och har efter det tillgång till alla andra domän

2.2 Fördelar med samlad inloggning

The Network Applications Consortium gjorde en undersökning bland stora företag och resultaten visade att användarna utför inloggningsaktiviteter i medeltal 44 timmar per år för att få tillgång till fyra program. Samma undersökning mätte innehållet på samtal som IT-supporten fick och det visade sig att 70 % av alla samtal var relaterade till lösenordsnollställning [3]. Så med andra ord så kommer en samlad inloggningslösning att öka på användarnas produktivitet och spara på IT-administrationskostnader.

En samlad inloggnings lösning kommer också att ge möjligheten för företag att stärka sin datasäkerhet med att implementera starkare lösenords regler. Även om företag inte vill ta detta steg så kan användarna ändå välja ett mera komplext lösenord eftersom de behöver endast komma ihåg ett ända lösenord.

Användarna behöver endast autentisera sig en gång per session för att få tillgång till alla domäner som han/hon behöver, dessutom så ökar produktiviteten på grund av denna lösning.

2.3 Nackdelar med samlad inloggning

Ett ofta mött argument mot samlad inloggning är att autentiseringsuppgifter för samlad inloggning ligger bakom en så kallad huvudnyckel. Alltså om någon får tag på samlade inloggningsautentiseringsdata för någon person, får denna tillgång till alla system som är säkrade av samlade inloggningssystem [3]. Ett annat scenario är att om en användare lämnar sin arbetsstation utan att logga ut sig från sin session så kan en obehörig person få tillgång till alla resurser den inloggade personen har tillgång till och möjligen orsaka obehörig intrång och skada. Dessa risker kunde reduceras med datasäkerhetsåtgärder som t.ex. automatisk utloggning av användare efter en viss tid samt att man väljer samlade inloggnings autentiseringsuppgifter vilka inte är kunskapsbaserade (lösenord är ett typiskt exempel) utan biometrisk baserade uppgifter (till exempel fingeravtryck) eller innehavs baserad (en kryptografisk markör eller ett smart kort). Autentisering med hjälp av flera användaruppgifter (alltså en kombination av kunskaps baserade uppgifter biometriska uppgifter och innehavs baserade) kommer ytterligare reducera risken [3].

3. Teknik som möjliggör samlad inloggning

3.1 Webbaserad samlad inloggning

Webbaserad samlad inloggning (Web SSO) används då man har webbresurser till förfogande. En användare som försöker ta kontakt med någon webbresurs via sin webbläsare kommer att bli dirigerade till en central autentiseringsserver var han/hon kommer att ge sina inloggningsuppgifter. Detta sker därför att de inte är autentiserade till domänen ifråga. Efter att de blivit autentiserade, kommer de att få en kaka vilket indikerar att de blivit autentiserade till domän. Efter detta så blir användaren dirigerade tillbaka till webbresursen var användaren sedan presenterar kakan, som den fick från autentiseringsservern. Webbresursen kommer att kommunicera med den centrala autentiseringsservern för att ta reda på om kakan är i kraft, om sessionen är ännu aktiv och vilka rättigheter användaren har. Efter

att användaren blivit autentiserad och auktoriserad, kommer han/hon att ha fullständig tillgång till alla webbresurser som han/hon har tillåtelse till [8, 9].

3.2 Legacy / Enterprise samlad inloggning

Legacy / Enterprise samlad inloggning är mycket likt webbaserad samlad inloggning för att båda är designade att hantera flera inloggningar till applikationer efter en autentiserings händelse [3, 4]. Funktionalitet för *Legacy / Enterprise* samlad inloggning inkluderar webbresurser och dessutom äldre applikationer inom ett företags interna nätverk.

3.3 Gemensam inloggning (Federated SSO)

Ett företag kan ha flera kunder eller affärspartners som behöver ha tillgång till en eller flera av företagets dataresurser. Låt oss säga att vi har en kund som vill logga in till företaget AB:s system. När kunden klickar på en länk till företag AB så gör kundens samlade inloggningssystem en säkerhetsförsäkran och vidarebefordrar denna. Företag AB:s samlade inloggningssystem kommer att ta emot denna försäkran, granska den och om den godkänns så kommer systemet att bevilja kunden tillgång till dataresurser utan att han/hon behöver logga in [8, 10].

För att få ett mera praktiskt exempel på gemensam inloggning så kan vi undersöka Shibboleth systemet. Shibboleth är ett samordnat identitetshanteringssystem. Den utgör en pålitlig metod för autentisering och auktorisering till en rad av olika leverantörer som alla är medlemmar i den gemensamma samordningen.

Det kan nämnas att Shibboleth systemet används också av Åbo Akademi. Denna följande förklaring finns bättre förklarat i [5]. Systemet består av fyra huvudsakliga enheter nämligen *Identity Provider* (IdP), *Service Provider* (SP), *Discovery Service* (DS) och användaren. IdP:n och SP:n kommunicerar med varandra i form av metadata filer, vanligen som XML-filer (*Extensible Markup Language*) för att mera noggrant kunna identifiera leverantören och vilka service den erbjuder. För att förstå hur Shibboleth login fungerar så förenklar vi detta med ett ”begäran för en resurs” process. Användaren begär efter en resurs via en SP

och den kan ha två möjliga åtkomsträttigheter, d.v.s. skyddad eller inte skyddad. I det första fallet så är användaren dirigerad till DS var användaren sedan med hjälp av ett grafiskt användargränssnitt väljer sin IdP. Om användarautentiseringen lyckas så kommer hemdomän IdP:n att konstruera en session och en koppling [5]. I största delen av fallen så kommunicerar IdP:n med användarens webbläsare och överlämnar kopplingen till SP:n som i sin tur använder den för att begära användarattribut från dess hem domän IdP:n. De begärda attributen från ett källsystem ges sedan till SP:n efter att den passerat ett rad huvudsteg. Det är viktigt att lägga märke till att IdP:n inte sparar några attribut för användaren. Den litar på att externa datalager tillhandhåller användare informationen för utgivning. Det finns möjligheten att datakällan innehåller ett antal flera attribut om användaren. Lanseringen av attributen till SP:n beror på SP:s politik vilket anger vilka attribut SP:n behöver om användaren för att framställa tillgångspolitik. IdP:n använder sig av attriubute-filter.xml-filen för att definiera vilka attribut som kommer att lanseras till vilka SP:n. Genom att känna till basen för begärda och mottagna attribut så bestämmer SP:n att antingen ge eller neka tillgång till det skyddade resursen/tjänsten. Ytterligare så är det viktigt att lägga märke till att enligt DS-kod så måste vi ha flera än en IdP för att installera och konfigurera DS:n för IdP:n. Detta är logiskt för att om vi har bara en IdP (vilket är fallet i största delen av fallen) finns det inget behov för en DS. En DS behövs då det finns flera än en IdP och att vi vill välja själv vilken vi vill använda [5].

4. Olika arkitekturer för samlad inloggning

Samlade inloggningsarkitekturer är delade i två huvudgrupper, de som hanterar bara en uppsättning av autentiseringsuppgifter och de som hanterar flera uppsättningar av autentiseringsuppgifter [15].

4.1 En uppsättning av autentiseringsuppgifter

De enklaste komplexa arkitekturerna för samlade inloggnings är de system som hanterar en uppsättning av autentiseringsuppgifter. Det finns två olika arkitekturer för samlad inloggning som hanterar en uppsättning av autentiseringsuppgifter, nämligen markörbaserad samlad inloggning och samlad inloggning baserad på *Public key infrastructure* (PKI).

Båda lösningar erbjuder samlad inloggning i en homogen omgivning, alltså en omgivning var alla enheter, applikationer och tjänster deltar i samma omgivning för samlad inloggning genom att använda samma kontonamngivningsformat och autentiseringsprotokoll [11].

4.1.1 Markörbaserad samlad inloggning

Figur 5 illustrerar hur en markörbaserad samlad inloggningslösning fungerar. Då användaren autentiserat sig till den primära autentiseringsauktoriteten så får användaren en mjukvarumarkör som kommer att sparas i användarens maskinminne. Denna markör kommer att användas för att bevisa användarens identitet för de sekundära autentiseringsauktoriteterna. För att validera markören så används kryptering i form av privata nycklar (symmetrisk kryptografi) mellan primära och sekundära autentiseringsauktoriteten. Denna kryptografi representerar förtroende mellan primära och sekundära autentiseringsdomänerna.

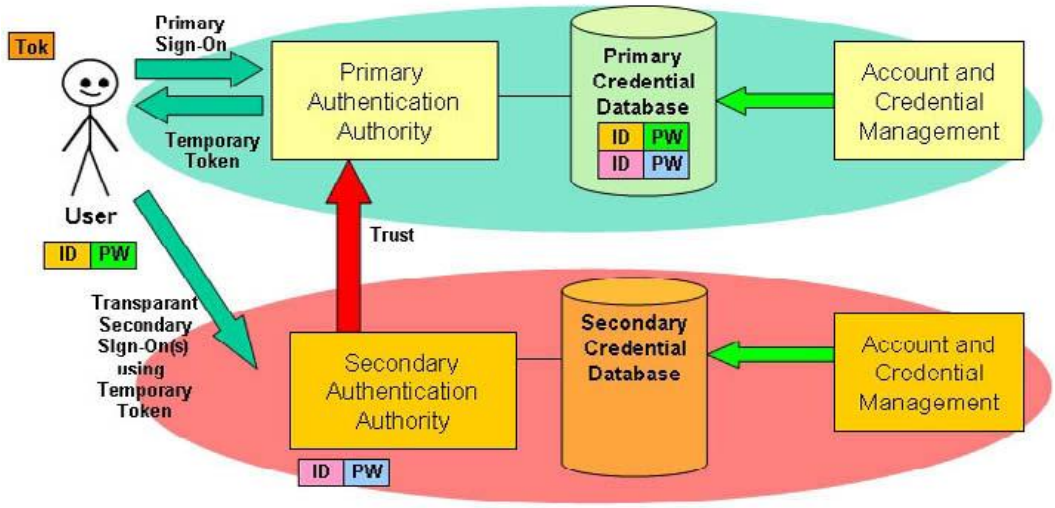


Fig. 3: [11] Markörbaserad samlad inloggning

Den kanske mest kända markörbaserade samlade inloggningslösningen är den som utvecklades av Massachusetts Institution of Technology (MIT), Kerberos. Den blev döpt efter den tre hövdade hunden som vaktar porten in till Hades i den grekiska mytologin. Kerberos används då en användare försöker få kontakt med en nätverkstjänst och tjänsten kräver autentisering. Efter att användaren autentiserat sig till Kerberos så får användaren en biljett från den och denna biljett innehåller information som länkar den till användaren. Användaren presenterar biljetten till nätverken för att få tillgång till tjänsten och nätverket analyserar biljetten för att verifiera användarens identitet och om användarverifieringen lyckas så får användaren tillgång till tjänsten [1, 2].

4.1.2 PKI-baserad samlad inloggning

Figur 6 illustrerar hur en arkitektur för samlad inloggning som baserar sig på PKI fungerar. Först registrera sig användaren till en autentiseringsauktoritet och detta kan ske på två olika sätt. Inom PKI-arkiteturen kallas autentiseringsauktoritetet för en certifikatutfärdare (*certificate authority*, CA) och användaren kan registrera sig dit eller sen kan användaren registerar sig till en av certifikatutfärdarens flera registraturfunktioner (*Registration Authority*, RA). Under registreringsprocessen sker det flera olika saker: användaren identifierar sig genom att använda en uppsättning av identifieringsvärden; en del av klientmjukvaran genererar ett asymmetriskt nyckelpar; och den öppna nyckeln kommer att presenteras åt certifikatutfärdaren eller registraturfunktionen för certifiering. Vid mottagning av användarens autentiseringsvärden och öppna nyckeln kommer certifikatutfärdaren

eller registraturfunktionen att verifiera användarens autentiseringsuppgifter. Om autentiseringsuppgifterna var korrekta kommer ett öppen nyckelcertifikat att generaras och detta certifikat kommer att skickas tillbaka till användaren var den kommer att sparas i användarens maskinminne. Dessa uppgifter används sedan för att producera en mjukvarumarkör som används för att bevisa användarens identitet till de sekundära autentiseringsauktoriteterna. Dessa mjukvarumarkörer liknar de som produceras i markörbaserad samlad inloggning.

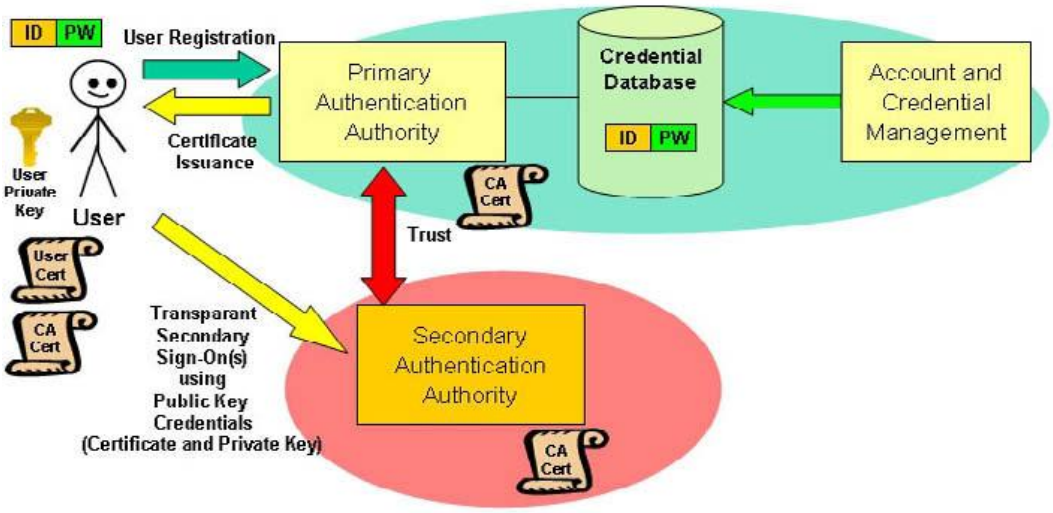


Fig. 4: [11] PKI-baserad samlad inloggning

Den största skillnaden mellan markörbaserad och PKI-baserad samlad inloggning är att PKI-lösningen använder asymmetrisk kryptografi för att verifiera användarens markör [2, 3].

4.2 Flera uppsättningar av autentiseringsuppgifter

Det finns tre stycken arkitekturer som kan hantera flera uppsättningar av autentiseringsuppgifter. Dessa är: Identitetsuppgiftssynkronisering, *Secure Client-side Credential Caching* och *Secure Server side Credential Cashing*, varav Identitetsuppgiftssynkroniseringslösning inte räknas som en riktigt samlad inloggningslösning. Dessa arkitekturer kan förverkliga samlad inloggning i en mer heterogent omgivning [3].

4.2.1 Secure Client-side Credential Caching samlad inloggning

Figur 3 illustrerar hur *Secure Client-side Credential Caching* samlad inloggning fungerar. I denna lösning autentiseras användaren genom att ange ett par av de primära autentiseringsvärdena. Efter autentiseringen får man tillgång till användarens identitetscache. Denna identitetscache finns sparad på användarens lokala maskins hårdskiva och innehåller användarens alla identitetsuppgifter som krävs för att man skall få tillgång till de andra resurserna. I fall vi inte har någon autentiseringsinfrastruktur, så används dessa primära autentiseringsuppgifter för att få tillgång till den lokala maskinen och dess säkerhetsdatabas och -resurser. Men ifall det finns en autentiseringsinfrastruktur används dessa primära autentiseringsuppgifter som domän autentiseringsuppgifter. Då användaren vill få tillgång till någon annan resurs eller applikation som kräver andra autentiseringsuppgifter kommer samlad inloggning programklientet att hämta de motsvarande autentiseringsuppgifterna från cachet och presenterar dessa till autentiseringsauktoritet. Om de presenterade uppgifterna är korrekta kommer användaren också bli automatiskt autentiserad till de andra resurserna. Därför måste de sekundära domänerna lita på den primära domänen [3], alltså måste en trust finnas mellan domänerna.

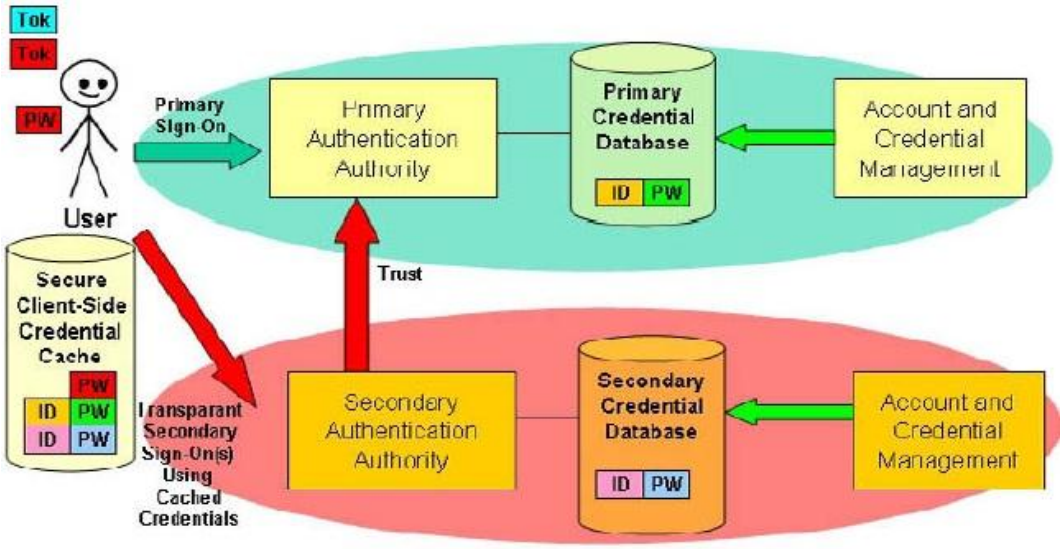


Fig. 5: [11] *Secure Client-side Credential Caching* samlad inloggning

I denna samlade inloggningsarkitektur är säker förvaring av identitetsuppgifterna det essentiella för att skydda tillgången till affärskritiska applikationer eller -data. Detta gäller speciellt bärbara klienter som måste ha en hög datasäkerhetsnivå [3].

Denna arkitektur har lite flexibilitet då all autentiseringsdata är sparad i klientautentiseringsuppgiftscachet i det lokala maskinet. Om en användare försöker autentisera sig genom en annan maskin så kommer det att orsaka autentiseringsproblem då autentiseringsuppgifter inte finns i den ifråga varande maskinen [4].

4.2.2 Secure Server side Credential Cashing samlad inloggning

Figur 4 illustrerar hur en *Secure Server side Credential Cashing* samlad inloggningslösning fungerar. I detta fall sparas autentiseringsuppgifterna i en central arkiv på en server istället för att dessa uppgifter skulle sparas på en lokal maskins säkerhetsdatabas. I denna arkitektur finns det ett så kallad huvudautentiseringsuppgiftsdatabas som innehåller en kartläggning av användarens primära- och sekundära autentiseringsuppgifter [3].

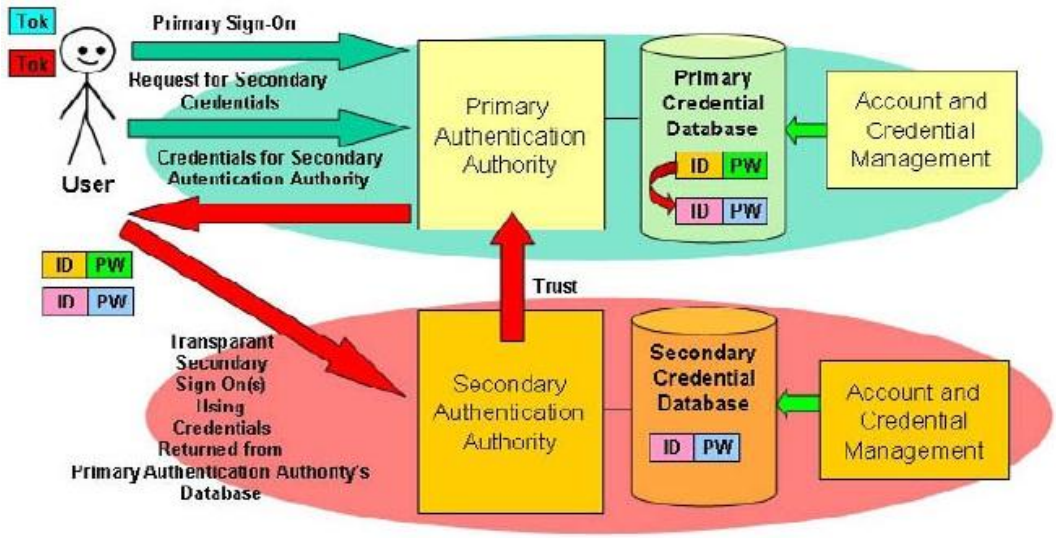


Fig. 6: [11] *Secure Server side Credential Cashing* samlad inloggning

I denna arkitetkur kommer användaren att först autentisera sig till den primära autentiseringsauktoriteten genom att använda sig av sina primära autentiseringsuppgifter och en förutbestämd autentiseringsprotokoll. Den samlade inloggningsmjukvaran kommer att antingen ge användaren en lista med tillgängliga applikationer eller överför en säkrad fil till användarens maskin som innehåller användarens sekundära autentiseringsuppgifter. Denna säkrade fil sparas hos den lokala klienten för resten av inloggningstiden och kommer att förstöras då användaren loggar ut sig från sessionen. Vid behov hämtas autentiseringsuppgifter automatiskt från denna säkrade fil och presenteras till autentiseringsauktoriteten. Om det igen skickas en lista på tillgängliga applikationer kommer den samlade inloggningslösningen att först kommunicera med den primära autentiseringsauktoriteten för att hämta de motsvarande autentiseringsuppgifterna, dessa är sedan vidarebefordrade till användaren på ett skyddat sätt. *Server side caching* medför högre säkerhet emedan autentiseringsuppgifterna sparas endast temporärt under inloggningstiden.

4.2.3 Identitetsuppgiftssynkronisering

Identitetsuppgiftssynkronisering anses inte vara en sann samlad inloggningslösning. Detta på grund av att användaren hamnar fortfarande att ange sina autentiseringsuppgifter till varje autentiseringsauktoritet. Figur 7 illustrerar hur en lösning för identitetsuppgiftssynkronisering fungerar.

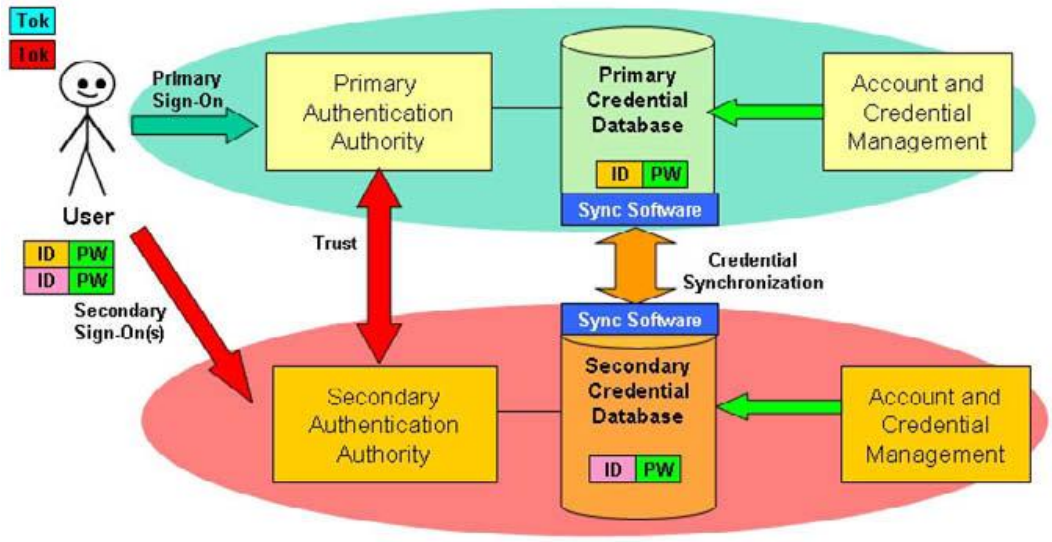


Fig. 7: [11] Identitetsuppgiftssynkronisering

En användare kan ha flera olika autentiseringsuppgifter, men på grund av Identitetsuppgiftssynkroniseringslösningen kommer autentiseringsuppgifterna att hållas identiska genom autentiseringsuppgiftssynkroniseringsmekanismen. Lösningar för Identitetsuppgiftssynkronisering använder sig av ett enda huvudidentitetsdatabas vilket kan gemensamt användas av administratörer för att uppdatera användarnas Identitetsuppgifter.

Teknologin bakom Identitetsuppgiftssynkronisering är inte lika lätt som Figur 7 får det att se ut. Ett problem är sätten identitetsuppgifterna sparas i identitetsuppgiftsdatabasen av de olika autentiseringstillhandhållare. Identitetsuppgifterna sparas oftast i hash format. Detta medför att det är omöjligt att lista ut lösenordet från denna format. Stor del av autentiseringstillhandhållare använder sig av olika hash format, därför går det inte att synkronisera databaserna rakt upp och ner. Identitetsuppgifterna kan endast uppdateras då de skapas eller uppdateras av användaren eller administratören[3, 6, 11].

4.3. Jämförelse för samlad inloggnings arkitekturer

Arkitektur	**Fördelar**	**Nackdelar**
Markör baserad samlad inloggning	Ett enda uppsättning av användarautentiserings-uppgifter underlättar användarnas och administrationens vardag	Autentiseringsinfrastruktur måste vara i en homogen omgivning.
		Använder sig av symmetrisk kryptografi.
PKI baserad samlad inloggning	En enda uppsättning av användarautentiserings-uppgifter underlättar användarnas och administrationens vardag.	Kan bara hantera en uppsättning av identitetsuppgifter.
	Använder sig av asymmetrisk kryptografi	Certifikatvalideringslogik är komplext vilket innebär att det krävs mycket bearbetning på klientsidan.
		Autentiseringsinfrastruktur delen måste vara i en homogen omgivning.
Secure Client Side Credential Caching samlad inloggning	Kan hantera flera olika uppsättningar av identitetsuppgifter	Kräver en skyddad klientsidig identitetuppgiftscache
	Kräver inte en homogen autentiseringsomgivning	Flera uppsättningar av identitetsuppgifter försvårar användares och administrationens liv
Secure Server side Credential Cashing samlad inloggning	Kan hantera flera olika uppsättningar av identitetsuppgifter	Kräver identitetsuppgifts-synkroniseringsmekanism
	Kräver inte en homogen autentiseringsomgivning.	Flera uppsättningar av identitetsuppgifter

		försvårar användares och administrationens vardag
Identitetsuppgiftssynkronisering	Kan hantera flera olika uppsättningar av identitetsuppgifter	Identitetsuppgifterna hålls identiska på olika plattformar
	Kräver inte en homogen autentiseringsomgivning	”Key to the kingdom” argumentet (om någon utomstående får reda på en lösenord så har den tillgång till hela systemet).
	Lättare att implementera än traditionell samlad inloggning	Flera uppsättningar av identitetsuppgifter försvårar användares och administrationens vardag.
		Kräver extra mjukvara på serverns infrastrukturdel.

5. Avslutning

Från en användares synvinkel är samlad inloggning en verkligen bra lösning. Det underlättar deras arbete eftersom de endast är tvungna att komma ihåg ett enda lösenord för att komma åt alla de dataresursser som de behöver under en arbetsdag.

I dessa dagar när datasäkerhet förspråkas så kan det lätt hända att en lösning blir oanvändbar på grund av att säkerhetsnivån är för hög. Det som man borde sträva efter är en balans mellan användarvänlighet och säkerhet. Samlade inloggningslösningar ger företagen möjligheten att kunna införa en stark lösenordspolicy, alltså att användarna skall välja ett komplext lösenord som de kommer ihåg. Samt att dessa lösenord skall kombineras med biometriska uppgifter om användaren för att få systemet så säkert som möjligt utan att riskera användarvänligheten.

Det jag lärt mig under denna avhandling är att det inte finns någon specifik lösning till ett företags krav på samlad inloggning. Det finns så många faktorer som måste tas i beaktande, några av dem innebär allmänna säkerhetsprinciper, typ av verksamhet, hur viktiga deras *legacy* och webbprogram är och kostnadsstrukturen för IT verksamheten. En grundlig undersökning bör utföras för varje faktor förrän man beslutar sig för något. Det kan även vara så att ett företag inte behöver en samlad inloggningslösning utan att de skulle klara sig med en lösning för Identitetsuppgiftssynkronisering, även om det inte räknas till en samlad inloggningslösning.

För tillfället är samlad inloggning den bästa lösningen vi har, fastän den inte är perfekt. Framtidsaspekterna är ljusa, i och med att tekniken går framåt vilket innebär att ny säkerhetsteknik kommer att införas till samlad inloggning.

Litteraturförteckning

[1]: Mark Ciampa, Ph.D., Security + Guide To Network Security Fundamentals, 4 ed., Course Technology CENGAGE Learning

[2]: William Stallings, Network Security Essentials Applications and standards, 4 ed., Pearson Education

[3]: De Clercq, Jan Grillenmeier, Guido, Windows Security Fundamentals, For Windows 2003 SP1 and R2, Digital Press

[4]: Si Xiong (12.02.2013) ”Web Single Sign-On System For WRL Company”, Master of Science Thesis, KTH, Tillgänglig: http://web.it.kth.se/~johanmon/theses/xiong.pdf

[5]: Zubair Ahmad Khattak, Suziah Sulaiman and Jamalul-lail Ab Manan, Security, Trust and Privacy (STP) Framework for Federated Single Sign-on Environment, November 2011, Digital Object Identifier: 10.1109/ICIMU.2011.612277 (Läst: 29.01.2013)

[6]: Andrej Volchkov**,** Revisiting Single Sign-On A Pragmatic Approach in a New Context, February 2001, Digital Object Identifier: 10.1109/6294.899932 (Läst: 28.01.2013)

[7]: The Open Group, Introduction to Single Sign-On, http://www.opengroup.org/security/sso/sso_intro.htm (Läst: 28.01.2013)

[8]: Frederick Chong, Identity and Access Management, Microsoft Corporation, July 2004, Tillgänglig: http://msdn.microsoft.com/en-us/library/aa480030.aspx

[9]: Bill Nelson, Single Sign-On Explained, December 17, 2011, Online blog, Tillgänglig: http://idmdude.com/2011/12/17/single-sign-on-explained/

[10]: Huntington Ventures Ltd., SSO Federation, 2006, Tillgänglig: http://www.authenticationworld.com/Single-Sign-On-Authentication/SSOFederation.html (Läst 26.02.2013)

[11]: Jan De Clercq, Security Consultant HP, Springer Berlin Heidelberg,2002, Single Sign-On Architectures, International Conference, InfraSec 2002 Bristol, UK, October 1–3, DOI: 10.1007/3-540-45831-X_4 (Läst: 28.01.2013)