Autentizace a autorizace v mobilních sítích

teze disertační práce

Autor: RNDr. Libor Dostálek

Školitel: Prof. Ing. Jiří Šafařík, CSc.

Konzultant specialista: Ing. Jiří Ledvina, CSc.

V Plzni dne: 28. dubna 2016

Podpis školitele:

1 Obsah

2 Úvod

V současné době existuje velké množství autentizačních metod. Přesto pro mobilní aplikace není k dispozici autentizační metoda, která by zcela vyhovovala mobilním aplikacím (viz kap. 3 Motivace).

Cílem práce je:

- Provést analýzu existujících autentizačních metod, zvolit metodu jejich hodnocení.
- Navrhnout nové autentizační metody pro mobilní aplikace (viz kap. 3 Motivace). *Během svého doktorského studia jsem již publikoval novou autentizační metodu* [1] [2]*. V současné době jsem navrhl další autentizační metodu, kterou uvádím v kapitole Dodatek, protože jsem metodu ještě nestihl publikovat.*
- Provést bezpečnostní analýzu komunikačních protokolů mobilních sítí a navrhnout způsob implementaci možných autentizačních metod za využití stávajících komunikačních protokolů mobilních sítí. Tj. navrhnout implementaci možných autentizačních metod bez nutnosti úpravy hardware stávajících mobilních zařízení.
- Na základě bezpečnostní analýzy porovnat navržené autentizační algoritmy.
- Publikovat řešení navržené v kapitole 1, včetně bezpečnostní analýzy tohoto řešení.
- Na základě bezpečnostní analýzy protokolů používaných v mobilních sítích navrhnout implementaci autentizace využívající více nezávislých ověřovatelů (kap. 0).
- Navrhnout poskytovatele identit založeného na bázi navržených autentizačních protokolů (viz kap. 4.3.1, 4.3.2).

Práce je členěna do následujících kapitol:

- Úvod – tato kapitola.
- Motivace: V této kapitole jsou shrnuty praktické problémy mobilních sítí, které mne motivovaly se tématem blíže zabývat.
- Současný stav vývoje: Tato kapitola obsahuje jednak obecné principy autentizace, a jednak volbu pokročilých autentizačních schémat, které byly dále zkoumány. Je zde rovněž zmiňováno, proč jiné metody zvoleny nebyly. Pro přehlednost práce jsou podrobnější části textu přesunuty do dodatků
- Cíle dizertační práce: Tato kapitola obsahuje předpokládané výstupy disertační práce a předpokládaný horizont dokončení dizertační práce. Jsou zde rovněž uvedeny výstupy, které již byly publikovány.
- Publikace autora: V této kapitole jsou jednak uvedeny publikace týkající se tématu disertační práce, a jednak i další publikace autora.
- Citovaná literatura.
- Dodatek: Nově navržená autentizační metoda (dosud nepublikováno).

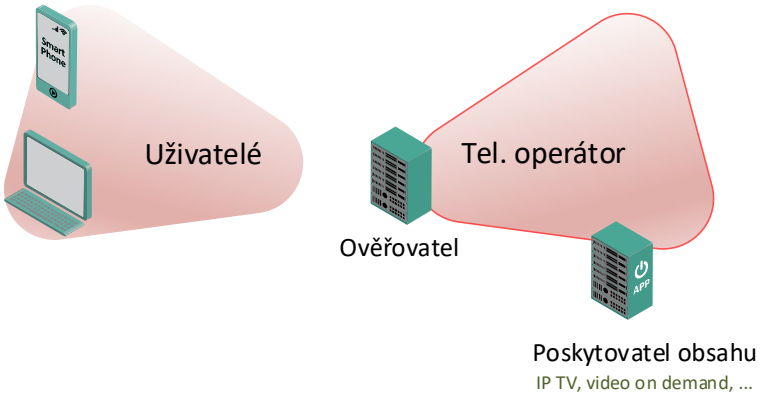
3 Motivace

V současné době se v 3G/4G mobilních sítích se používá autentizace pomocí mechanismu AKA [3], jak v UMTS [3], tak v LTE [4], tak i v IMS (VoLTE) [5]. Pomocí schématu AKA se uživatel autentizuje na celou dobu od přihlášení uživatele do sítě až do jeho odhlášení (resp. vypnutí zařízení). Pokud síť potřebuje znovu autentizovat mobilní zařízení přihlášeného uživatele (např. přechod do roaming), může provést autentizaci bez zásahu uživatele. Tj. útočník má šanci zaútočit kdykoliv mezi přihlášením a odhlášením uživatele, pokud např. pomocí zlomyslného kódu ovládne uživatelovo zařízení.

Schéma AKA může být využíváno nejenom k autentizaci pro hlasové (multimediální) služby, ale i k autentizaci do internetových aplikací [6] na bázi protokolu HTTP [6]. Avšak jeho využití pro internetové aplikace je omezené na úzký okruh aplikací. Mobilní webové aplikace zpravidla nevyužívají AKA autentizaci - používají autentizace heslem. Protokoly autentizace heslem jsou zdokonalovány tak, aby odolávaly známým útokům (např. [7] [8] [9]).

V současné době nasazovaný *Internet Multimedia Subsystem* (IMS) [10] bude služby telekomunikačních operátorů řešit jako aplikace. Aplikace budou zajišťovat nejen hlasovou komunikaci (obecně multimediální komunikaci), ale i další aplikační služby. Aplikační služby budou moci být poskytovány i třetími stranami. Příklady aplikačních služeb mohou být např. videokonference, ale např. i *video on demand* atp.

V případě poskytování aplikačních služeb se jedná o obdobnou situaci, která byla před lety na Internetu, kdy poskytování obsahu bylo v režii poskytovatelů připojení. Teprve



Obr. 3.1 Poskytovatel obsahu využívá autentizaci operátora sítě

v okamžiku, kdy obsah začaly poskytovat 3. strany (poskytovatelé obsahu), se Internet rozvinul do současných rozměrů.

Autentizace do aplikací přitom může být v současné době řešena jednou z následujících možností:

- Autentizace si řeší sám poskytovatel služby (bez účasti operátora), např. pomocí jména a hesla.

- K autentizaci se využijí prostředky pro autentizaci účastníka mobilní sítě (tj. např. USIM karta). Jedná se sice o silnou autentizaci, ale ta je pod výhradní kontrolou operátora. Vlastník aplikace (3. strana) si může dělat správu svých uživatelů jen velice omezeně – nemá správu uživatelů zcela pod svou kontrolou.

Motivací je snaha navrhnout autentizační algoritmy, které by využily silnou autentizaci účastníka mobilní sítě a přitom umožnily poskytovatelům obsahu mít správu svých uživatelů ve své moci.

3.1 Druhý autentizační faktor

Je třeba rozvést termín „dvou (resp. více) faktorová“ autentizace (blíže viz kapitola 4.2). Jak AKA mechanismus, tak i sofistikovanější autentizace pomocí hesla (např. [11], [8], [9]) jsou dvou faktorovými autentizacemi:

- AKA mechanismus využívá USIM/ISIM čipovou kartu a PIN.
- Sofistikovanější autentizace pomocí hesla (např. [11], [8], [9]) využívají heslo a kryptografický materiál uložený na nějakém datovém nosiči (opět např. na čipové kartě).

Problémem ale je, že oba autentizační faktory jsou na straně aplikace (ověřovatele) v moci téže osoby. Což má následující nevýhody:

- V případě AKA autentizace je USIM/ISIM čipová karta poskytována telekomunikačním operátorem. Což je těžko přijatelné pro poskytovatele aplikací, protože správa jeho uživatelů je v moci telekomunikačního operátora.
- V případě sofistikovanější autentizace pomocí hesla (např. [11], [8], [9]) by z bezpečnostního hlediska nevadilo, že uživatel využívá heslo i např. čipovou kartu obdrženou od poskytovatele aplikace. Avšak z technického hlediska je problém, jak tuto aplikačně závislou čipovou kartu využívat uživatelovým zařízením. Z praxe víme, že technické problémy spojené s takovou implementací často přinášejí uživatelům těžké problémy. Navíc každý poskytovatel aplikace by využíval jiné čipové karty.

3.2 Cíl

Cílem je navrhnout takový autentizační algoritmus pro mobilní webové aplikace, který propojí AKA schéma se silnou autentizací heslem. Pokud možno takový, který umožní i autorizaci dat. Takovéto schéma bude užitečné zejména pro nové aplikace v nových mobilních sítích, kdy uživatel je neustále připojen k internetu (což vyplývá z podstaty těchto sítí).

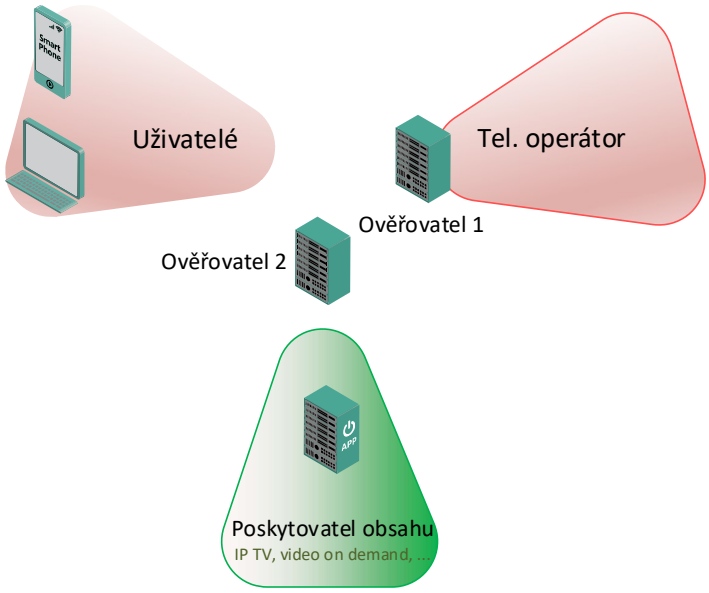
Dalším důvodem k hledání nového algoritmu je již zmíněná skutečnost, že tajemství pro autentizaci AKA schématem spravuje operátor sítě. Pro nezávislé poskytovatele obsahu to znamená, že správa uživatelů je plně v moci telekomunikačního operátora (Obr. 3.1).

Cílem je, aby k autentizaci byl použit druhý autentizační faktor (např. druhý ověřovatel), který by si spravoval poskytovatel aplikace (Obr. 3.2).

Kombinací AKA schématu a algoritmu silné autentizace heslem získáme více faktorovou autentizaci, která pevně spojí držitele USIM/ISIM s jeho heslem do aplikace. Heslo aplikace přitom bude ve správě poskytovatele aplikace.

3.3 Námitky

Při publikování navržené autentizační metody [1] [2] jsem se setkal s námitkou, že se nejedná o novou myšlenku, protože existuje celá řada dvou i více faktorových autentizačních metod, které je možné využít i v mobilních aplikacích (např. [11], [8], [9]). To je pravda, ale tyto dvou faktorové autentizace zásadně využívají jen jednoho ověřovatele. Tj. nezavazují autentizaci uživatele s USIM/ISIM kartou – neřeší problém uvedený v kap. 3.1.



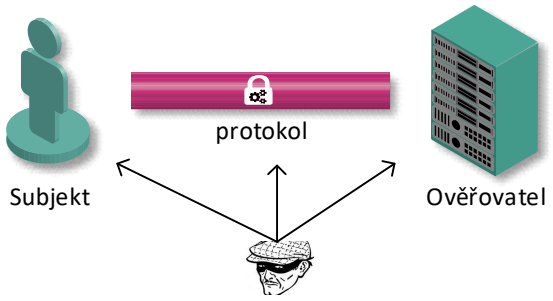
Obr. 3.2 Druhý autentizační faktor spravovaný poskytovatelem aplikace

4 Současný stav vývoje

Autentizace je proces ověření identity subjektu. Tento proces provádí ověřovatel, který vydává záruku, že subjekt má deklarovanou identitu (Obr. 4.1). Kvalita této záruky závisí na konkrétním procesu autentizace.

Rozlišujeme autentizaci entity a autentizaci zprávy. Rozdíl je v časovém hledisku. Autentizace zprávy (např. pomocí elektronického podpisu) nedává záruku o tom, kdy byla zpráva vytvořena[1]. Naopak autentizace entity zahrnuje doložení identity žadatele zpravidla prostřednictvím aktuální komunikace s ověřovatelem.

Příkladem procesu autentizace je proces, kterým se uživatel pomocí uživatelského jména a hesla přihlašuje do aplikace.



Obr. 4.1 Autentizace

Vedlejším efektem procesu autentizace může být skutečnost, že během autentizace entity bude rovněž vygenerován kryptografický materiál, který bude sloužit k zabezpečení následné komunikace.

4.1 Metody autentizace

Metody autentizace lze rozdělit do následujících kategorií:

1. Subjekt něco ví – např. autentizační faktory: heslo, soukromý nebo tajný klíč, sdílené tajemství atp.
2. Subjekt něco má – např. autentizační faktory: čipová karta, kalkulátor pro generování jednorázových hesel atp.
3. Subjekt něčím je – např. autentizační faktory: otisk prstů, dynamický biometrický podpis, tvar krevního řečiště ruky atp. V poslední době se pak hovoří i o tzv. digitální stopě.

4.1.1 Kategorie „Něco ví“

Pro kategorii „něco ví“ Máme následující typy autentizačních metod:

[1] V případě elektronického podpisu se tato informace dodatečně k elektronickému podpisu přikládá zpravidla formou časového razítka.

- Autentizace na základě hesla.
- Autentizace pomocí dialogu.
- „Zero-knowledge“ autentizace.

Autentizace na základě hesla

Heslo je pro uživatele zapamatovatelný řetězec znaků, který je platný po jistou dobu. Obecně se autentizace heslem považuje za slabou. Existují ale i slabší autentizační metody, jako je např. autentizace na základě IP-adresy (viz též digitální stopa kap. 4.1.3).

Vedle hesel máme jednorázová hesla, tj. hesla, která je možné použít pouze jednou. Pro vytváření jednorázových hesel existuje celá řada algoritmů. Od prostého seznamu jednorázových hesel, přes algoritmy založené na sdíleném tajemství mezi subjektem a ověřovatelem až např. po tzv. Lamportovo schéma [12]. Schémata pro generování jednorázových hesel už ale zpravidla řadíme do autentizace pomocí dialogu.

Autentizace pomocí dialogu

Dialog se může např. skládat za dvou kroků: výzvy a odpovědi. Výzva obsahuje řetězec obsahující náhodné číslo, pořadové číslo autentizace, čas atp. V odpovědi pak nalezneme řetězec z výzvy, na který byla aplikována symetrická šifra, asymetrická šifra nebo jednocestná funkce. Aby autentizace mohla proběhnout, tak předem musí být mezi subjektem a ověřovatelem vyměněny tajné informace: např. kryptografické klíče, resp. sdílená tajemství. Tyto tajné informace se pak použijí např. jako šifrovací klíče, kterými se šifruje výzva. V případě jednocestné funkce se tajná informace sřetězí s výzvou před tím, než se na výzvu aplikuje jednocestná funkce.

Autentizace pomocí dialogu se někdy označuje jako silná (*strong*) v protikladu s autentizací heslem.

Pro svou práci jsem hledal silnou autentizaci (tj. autentizaci pomocí dialogu), která používá heslo. Při rešerši autentizačních schémat jsem zjistil, že tato schémata lze rozdělit do dvou skupin v závislosti na tom, zda na straně autentizovaného subjektu používají nebo nepoužívají datový nosič pro uložení dalšího kryptografického materiálu. Jako datový nosič se zpravidla používá čipová karta, proto v názvech schémat se často vyskytuje sousloví „čipová karta“.

V literatuře se tato schémata označují jako:

- Autentizace pomocí hesla „bez čipové karty“. Sem patří již zmíněné Lamportovo schéma [12], ale byla publikována i další schémata, např. [7]. Schémata z této skupiny schémat „bez čipové karty“ jsou dnes obecně brána jako slabá a dále se jimi již nezabývám.
- Autentizace pomocí hesla „s čipovou kartou“. Místo sousloví „čipová karta“ budu raději používat termín „datový nosič“, aby nedošlo k záměně s čipovými kartami USIM/ISIM využívanými mobilními zařízeními.

Nadále se budu věnovat schématům „s čipovou kartou“ (resp. „s datovým nosičem“). Publikováno bylo několik schémat. U některých schémat po jejich publikaci následovalo publikování jejich slabin, zpravidla doplněných změnou schématu (resp. navržením schématu nového) tak, aby byla slabina odstraněna.

Mým cílem bylo najít pokročilé autentizační schéma. Požadoval jsem, aby autentizační schéma minimálně umožňovalo:

- Oboustrannou autentizaci.
- Změnu hesla.
- Nastavení hesla v případě zapomenutí hesla.

- Odolnost schématu proti následujícím útokům:
    - Vylákání hesla.
    - Odposlechnutí hesla.
    - Uhodnutí hesla.
    - Útoky na synchronizaci času. Některé autentizační mechanismy využívají aktuální čas. Jelikož jsou známy útoky proti tomuto způsobu autentizace, vyžadujeme nezávislost na aktuálním čase.
    - Útoky na synchronizaci komunikace (např. na autentizační dialog). Subjekt ani ověřovatel nesmějí být de-synchronizovány tak, aby si každý myslel, že používá jiné sdílené tajemství.

Těmto požadavkům vyhovuje řada autentizačních protokolů. Jak již bylo zmíněno, tak většinou po jejich publikaci následovalo publikování jejich slabin a návrh dalších protokolů odolných proti zjištěným slabinám. Výsledkem této diskuse jsou pak mj. dva protokoly:

- *Secure Hash-Based Password Authentication Protocol Using Smartcards* [8]. Protokol založený na hešovacích funkcích. Tomuto schématu předcházela publikovaná schémata, u kterých se ukázaly slabiny (např. [13] a [11]).
- *Robust Two-Factor Authentication and Key Agreement Preserving User Privacy* (Robustní dvou-faktorová autentizace) [9] (viz též kap. 7.1). Jedná se o protokol založený na eliptických křivkách. Tomuto autentizačnímu schématu rovněž předcházela diskuse (viz např. [14]).

Robustní dvou-faktorová autentizace je novější schéma (2014). Toto schéma přišlo ještě s dalšími bezpečnostními požadavky:

- Odvolání datového nosiče („čipové karty“). Tj. v případě, ztráty čipové karty či zrušení platnosti čipové se musí útočníkovi zbránit zneužít čipovou kartu.
- Anonymita autentizovaného subjektu (*User anonymity*). Tj. třetí osoba sledující autentizační dialog nezjistí identitu subjektu.
- Nevystopovatelnost autentizovaného subjektu. Tj. třetí osoba sledující autentizační dialog nezjistí, kdy se jaký subjekt autentizuje.
- Generování kryptografického materiálu pro zabezpečení následné komunikace za podmínek:
    - *Session key agreement* – Během autentizace dojde k ustavování relace, kdy se obě strany dohodnou na kryptografických klíčích relace, které budou známy pouze subjektu a ověřovateli a budou využívány jen pod dobu relace.
    - *Perfect forward secrecy* - Útočník se nemůže dostat k datům relace, i když v budoucnu bude kompromitován některý ze soukromých klíčů (subjektu nebo ověřovatele), kterými se provádí počáteční autentizace.

- *Forward and backward secrecy* - Vyzrazení klíče relace nepomůže k získání klíčů budoucích nebo minulých relací.
- *Key freshness* - Ani jedna ze stran nemůže předurčit sdílený klíč relace před zřízením relace.

V neposlední se zabývám protokolem AKA ([3]), který využívá 3. a 4. generace mobilních telefonů.

Problém je, jak porovnat jednotlivá autentizační schémata (autentizace pomocí dialogu). V zásadě se používá dvojí porovnání:

- Vezme se seznam známých útoků na autentizaci a zjišťuje se, zda jsou daná autentizační schémata odolná proti těmto útokům. Zpravidla se bezpečnostní vlastnosti jednotlivých schémat porovnají v tabulce (např. Tabulka 1).
- Porovnává se výpočetní náročnost na autentizaci pomocí jednotlivých schémat. Zpravidla se zjišťují počty: asymetrických operací, symetrických operací a aplikací jednocestných funkcí (např. Tabulka 2).

Tabulka 1 Příklad porovnání schémat z hlediska bezpečnostních vlastností „něco ví“

	Bezpečnostní vlastnost	Odolné proti (není slabinou)			
		Heslo	*Secure Hash-Based Password Authentication Protocol Using Smartcards* [13]	*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy (kap. 1)*	AKA
1	Oboustranná autentizace	Ne	Ano	Ano	Ano
2	Změnu hesla součástí schématu	Ne	Ano	Ano	Ano
3	Nastavení hesla v případě zapomenutí hesla (jako součást schématu)	Ne	Ano	Ano	Ano
4	Vylákání hesla	Ne	Ano	Ano	Ano
5	Odposlechnutí hesla	Ne	Ano	Ano	Ano
6	Uhodnutí hesla	Ne	Ano	Ano	Ano
7	Útoky na synchronizaci času	Ano	Ano	Ano	Ano
8	Útoky na synchronizaci komunikace	Ne	Ano	Ano	Ano
9	Odvolání datového nosiče	-	Ano	Ano	Ano
10	Anonymita autentizovaného subjektu	Ne	Ano	Ano	Ne
11	Nevystopovatelnost	Ne	Ano	Ano	Ne
12	*Session key agreement*	Ne	Ne	Ano	Ano
13	*Perfect forward secrecy*	-	-	Ano	Ano

14	*Forward and backward secrecy*	-	-	Ano	Ano
15	*Key freshness*	-	-	Ano	Ano

Tabulka 2 Příklad porovnání schémat pomocí výpočetní náročnosti

		Heslo	*Secure Hash-Based Password Authentication Protocol Using Smartcards* ([13])	Robust Two-Factor Authentication and Key Agreement Preserving User Privacy *(kap. 1)*
1	Výpočty na straně uživatele	1h	3h, 1A	4h, 1H, 2EC
2	Výpočty na straně serveru	1h	4h,1A	3h, 1H, 2S, 2EC
3	Počet vyměněných zpráv během autentizace	2	3	3

Kde :

h	Označuje čas spotřebovaný hešovací funkcí
H	Čas strávený hešovací funkce, jejímž výsledkem je bod na eliptické křivce (*map-to-point on elliptic curve hash function*)
S	Čas strávený symetrickým šifrováním/dešifrováním
EC	Čas strávený algoritmem eliptických křivek
A	Čas strávený asymetrickým algoritmem (např. RSA)

Autentizace Zero Knowledge

Autentizace pomocí hesla nebo dialogu je založena na znalosti tajné informace (heslo, sdílené tajemství atp.). Jelikož tajnou informaci zná jen subjekt a ověřovatel, předpokládá se, že to je dostatečný důkaz o pravosti klienta. Slabinou těchto metod je skutečnost, že se tajná informace nějakým způsobem během autentizace poodhaluje, což může být příležitost pro útočníka.

Zero Knowledge schémata vycházejí z předpokladu, že subjekt má znalost nějakého složitého problému (je to jeho tajemství). Autentizace pak probíhá pomocí předvedení znalosti řešení tohoto složitého problému (např. NP problému). Výsledkem autentizace je pak jen jednobitová informace autentizován/neautentizován. To je sice z hlediska bezpečnosti velice zajímavé, protože se nepoodhaluje tajemství, ale tyto algoritmy negenerují kryptografický materiál pro zabezpečení následné komunikace, proto je dále již nebudu rozpracovávat.

4.1.2 Kategorie „Něco má“

Autentizační kategorie „Něco má“ může mít v reálném světě nejrůznější podoby – např. plastikový průkaz ke vstupu. V mobilních sítích to může být:

- Čipová karta nebo její obdoba. Tj. jednočipový počítač s uloženým kryptografickým materiálem sloužícím pro autentizaci osob (tj. zařízení pro uložení osobních autentizačních aktiv). Toto zařízení během autentizace elektronicky komunikuje s ověřovatelem. Přístup k osobním aktivům (kryptografickému materiálu) je zde chráněn:
    - Jedním nebo více PINy v případě přístupu osoby (držitele).
    - Mechanismem *Secure Messaging* v případě přístupu aplikací bez zásahu uživatele (držitele).
- Autentizační kalkulátor je rovněž jednočipový počítač s uloženým kryptografickým materiálem sloužícím pro autentizaci osob (tj. zařízení pro uložení osobních autentizačních aktiv), ale zpravidla elektronicky nekomunikuje s ověřovatelem, ale informaci zobrazí na displeji. Držitel pak informaci opíše a předá ověřovateli.
- HSM (*Host Security Module*, někdy též *Hardware Security Module*) je výkonný počítač sloužící pro uložení aktiv systému (např. serveru). Přímo elektronicky komunikuje se systémem.
- Mobilní telefon.

Toto dělení je dnes považováno za historické. Organizace GlobalPlatform abstrahovala od konkrétního fyzického provedení a definovala tzv. Bezpečný prvek (*Secure Element*) pro uchovávání osobních kryptografických aktiv [15]. Prakticky je míněn specializovaný jednočipový mikroprocesor určený pro bezpečné uchovávání kryptografických dat a bezpečné provádění operací s nimi. Uvedené operace se provádějí v tzv. *Trusted Execution Environment* (TEE) [16].

Bezpečný prvek může být realizován jako součást čipové karty (USIM, ISIM, SD atp.) nebo např. jako čip integrovaný na základní desce mobilního zařízení atp. Na bezpečný element se z hlediska bezpečnosti v podstatě díváme obdobně jako na HSM.

Závěrem lze tedy říci, že osobní autentizační aktiva mohou být uložena:

- Na datovém nosiči bez ochrany (resp. se slabou ochranou).
- V bezpečném prvku (*Secure Element*).
- V HSM modulu.

V případě porovnávání jednotlivých metod bereme v úvahu následující bezpečnostní vlastnosti (závisí též na konkrétní implementaci):

- Zařízení fyzicky uchovává kryptografický materiál (a aplikace jej využívá).
- Přístup ke kryptografickému materiálu pomocí hesla nebo PIN.

- Kryptografický materiál neopouští zařízení (je neexportovatelný).
- Zařízení je fyzicky chráněno proti neoprávněnému přístupu.

4.1.3 Kategorie „Něčím je“

Touto autentizační kategorií se zpravidla myslí autentizační faktory založené na biometrických vlastnostech subjektu, tj. ověření identity osoby na základě měřitelných fyziologických nebo behaviorálních charakteristik, jedinečných a relativně neměnných pro subjekt.

Tabulka 3 Některé další veličiny biometrické autentizace

FMR (*False Match Rate*) – pravděpodobnost, že daný vzorek bude nesprávně označen za shodný s některým náhodně vybraným referenčním vzorkem jiné osoby.
FNMR (*False Non-Match Rate*) – pravděpodobnost, že vzorek určité osoby bude nesprávně označen za odlišný od referenčního vzorku stejné osoby.
FAR (*False Acceptance Rate*) – míra nesprávných přijetí; pravděpodobnost, že neautorizovaná osoba získá přístup do systému – je nesprávně označena jako některý oprávněný uživatel.
FRR (*False Rejection Rate*) – míra nesprávných odmítnutí; pravděpodobnost, že oprávněnému uživateli je zamítnut přístup do systému – biometrický vzorek uživatele není označen za shodný s jeho referenčním vzorkem, případně s žádným vzorkem v databázi.
FTE (*Failure To Enrol Rate*) – pravděpodobnost, že daná osoba není schopna úspěšné registrace do systému. Příčinou může být chybějící orgán, zranění, velká variabilita mezi jednotlivými vzorky předkládanými při registraci, nemožnost poskytnout vzorek dostatečné kvality, apod.
FTA (*Failure To Acquire Rate*) – pravděpodobnost, že systém není schopen během měření získat vzorek dostatečné kvality. Chyba může být způsobena nesprávnou prezentací biometriky snímači, zraněním, nedostatečnou kvalitou získaného vzorku nebo i vlivy okolního prostředí.

Konkrétní biometrická charakteristika se subjektu nejprve sejme a vytvoří se tzv. vzor. Autentizace pak probíhá na zjišťování korelace aktuálních charakteristik subjektu s charakteristikami uloženými ve vzoru. Vedle korelace se sledující další veličiny (viz Tabulka 3 [17]).

Základní nevýhodou biometrických charakteristik je, že je v případě zneužití nelze odvolat a následně změnit. Např. pokud útočník získá dynamický biometrický podpis subjektu, pak subjekt již nikdy nemůže dynamický biometrický podpis používat, aniž by nehrozilo jeho zneužití.[2]

[2] V případě podpisu je možné k podpisu připojovat např. číslici, obrázek apod. Tím se de facto odvolá předchozí podpis bez číslice či obrázku.

Digitální stopa (*Digital Footprint*) má obdobné vlastnosti jako biometrické charakteristiky. Jedná se zejména o sledování metadat, která používáme při komunikaci nebo která po sobě zanecháváme.

Součástí digitální stopy může být:

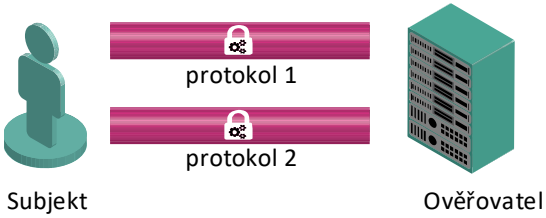
- IP adresa, resp. autonomní systém, ze kterého IP adresa je.
- Metadata aplikačního protokolu. V protokolu HTTP to může být např.: hlavička User-Agent, hlavičky Acceept*, navštívená URL, cookies atd.
- Chování klienta v aplikaci (např. počet autentizací za den, obvyklá velikost transakcí atd.).
- Použité autentizační metody.
- Metadata přejatá z autentizace pomocí federaci identit.

Pokud si budeme tato metadata o subjektu (digitální stopu) ukládat, pak můžeme zjišťovat korelaci aktuálních metadat s uloženými. Situace zde není tak jednoznačná, protože subjekt může přistupovat z různých systémů či může cestovat. Subjekt si můžeme vytvořit více profilů subjektu (mobilní, osobní počítač atp.) – obdobně v případě otisků prstů může snímat otisky více prstů. Digitální stopa se v praxi hojně využívá např. v případě cílené reklamy.

Digitální stopa má oproti biometrickým charakteristikám výhodu v tom, že ji lze změnit. Nevýhoda spočívá v problému sporné legálnosti sledování osobních údajů.

4.2 Více faktorová autentizace

Více faktorová autentizace[3] znamená, že pro autentizaci se použije dvou nebo více autentizačních faktorů (např. dva různé protokoly - Obr. 4.2).



Obr. 4.2 Více faktorová autentizace

Přitom je důležité, aby byly použity dva odlišné autentizační faktory. Např. použití dvou hesel za sebou příliš kvalitu autentizace nezlepší. Autentizační faktory se mohou lišit:

- Různým kryptografickým materiálem.
- Různým autentizačním schématem.
- Různým komunikačním protokolem.
- Různým komunikačním kanálem.
- Různým ověřovatelem.

Důležité rovněž je, aby autentizační faktory byly provázané. Pokud nejsou, pak se útočníkovi ulehčuje práce, neboť útočník se nejprve může věnovat zlomení jednoho autentizačního faktoru a pak druhého. Ne však vždy toho lze prakticky dosáhnout. Např. pokud je již subjekt autentizován (např. si přinesl autentizace z aplikace Facebook) a ukáže se, že pro danou operaci je nutná silnější autentizace (např. čipovou kartou), pak se zpravidla re-autentizuje jen silnějším schématem (čipovou kartou), které je nezávislé na původní autentizaci.

4.3 Federace identit

Byla-li identita ověřena jedním ověřovatelem, pak je otázkou, zda by i jiný ověřovatel mohl tomuto ověření věřit. Tj. zda by ověřovatel akceptoval ověření subjektu od jiného ověřovatele bez toho, aby sám provedl ověření.

Jedná se o standardní požadavek, který už řešil protokol Kerberos (založený na schématu Needham-Schroeder [18]). První verze protokolu Kerberos byla publikována v roce 1987 (aktuální verze [19]). Tento protokol používá pro subjekt termín principál, pro ověřovatele termín KDC (*Key Distribution Center*). KDC ověřuje identitu principálů v rámci své říše (*Realm*). Výsledkem ověření je systém lístků (*ticket*), pomocí kterých lze přistupovat ke službám v rámci říše.

[3] Někdy se též používá termín vícesložková autentizace

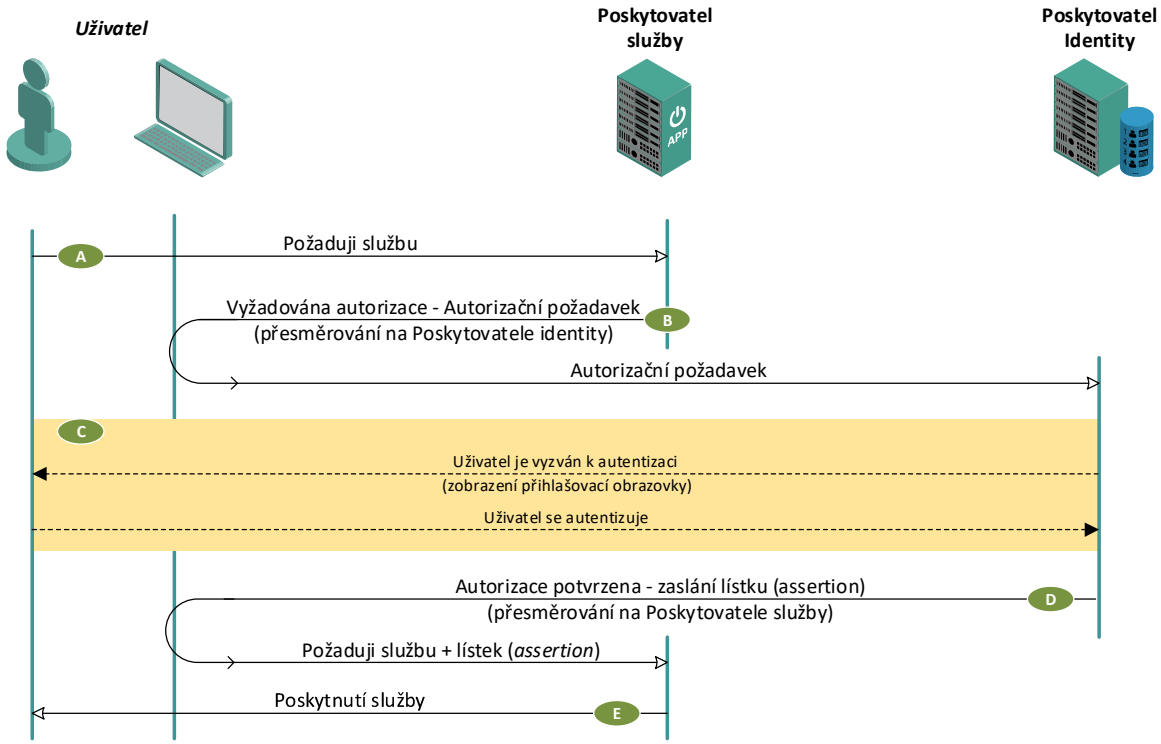
Protokol Kerberos řeší i problém důvěry mezi říšemi, tj. federaci - neboli problém jak se lístkem vydaným v jedné říši prokázat v jiné říši.

V současné době se, pro federaci identit (kromě protokolu Kerberos) používají dva hlavní standardy:

- *Security Assertion Markup Language* (SAML) [20] – nyní ve verzi 2.
- *Open Authentication* (OAuth) [21] [22] [23] – nyní ve verzi 2.

4.3.1 SAML

SAML (*Security Assertion Markup Language* [20]) řeší problém federace tak, že odděluje poskytovatele služby (tj. poskytovatele zdroje informace o kterou má subjekt zájem) a poskytovatele identity. Poskytovatel identity provádí autentizaci subjektu. Výsledkem autentizace je vydání lístku[4] (*Assertion*), na základě kterého poskytovatel služby poskytne/neposkytne příslušný zdroj. Federace spočívá v tom, že Poskytovatel identit poskytuje lístky více poskytovatelům služeb (Obr. 4.3).



Obr. 4.3 Komunikační schéma SAML 2.0

SAML sám je jen manipulační jazyk, který popisuje samotný lístek (*Assertion*). Tj. jak dialog autentizace, tak mechanismus přesměrování[5] znázorněný na Obr. 4.3 jsou mimo specifikaci tohoto standardu. Tj. závisí na konkrétní implementaci, ale obecně se předpokládá využití protokolu HTTP. Cože je případ díle popisovaného OAuth 2.0.

4.3.2 JWT

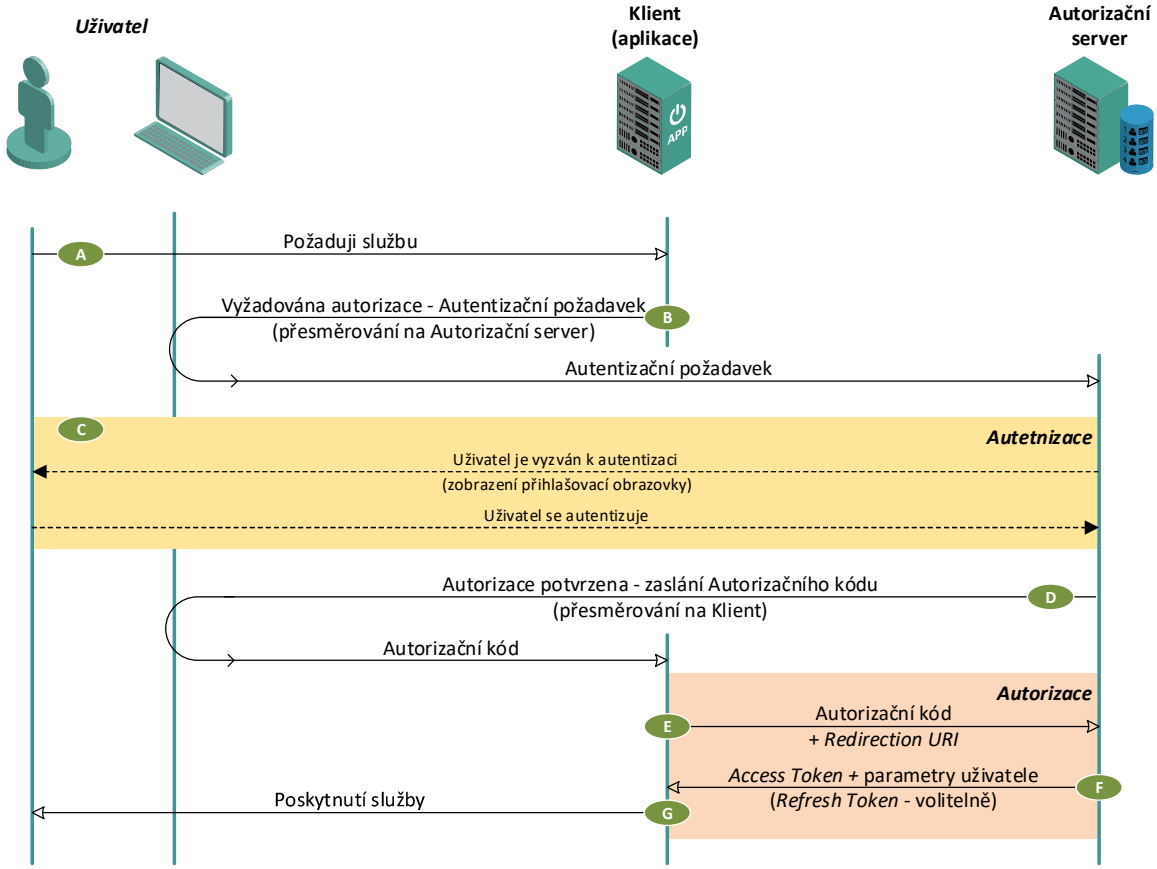
Manipulační jazyk SAML je velice obecný, ale díky své obecnosti je jednak složitý a jednak výsledný lístek příliš dlouhý, což někdy vyvolávalo technické obtíže. Autentizační informace se proto dnes častěji nepopisují ve tvaru SAML, ale pomocí *JavaScript Object Notation* (JSON). Vznikl tak standard JSON Web Token (JWT) [24].

[4] Někdy se též používají termíny oprávnění, tvrzení, token atp.

[5] Zpravidla se využívá mechanismus přesměrování protokolu HTTP

4.3.3 OAuth 2.0

Jak SAML, tak i JWT popisují jen lístek, který vydává poskytovatel identity subjektu, aby se jím prokázal poskytovateli služby. Protokol, kterým dojde k této komunikaci, je mimo tyto standardy (tj. není součástí těchto standardů).



Obrázek 4.4 Dialog protokolu OAuth 2.0

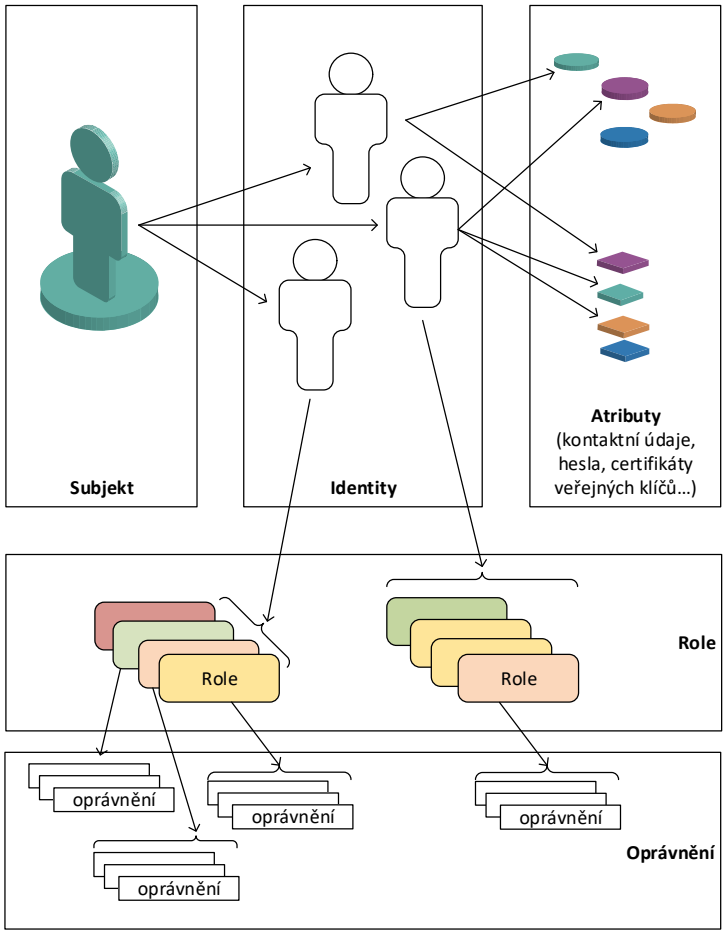
OAuth 2.0 [21] [22] [23] je protokol, který tento problém řeší, tj. popisuje tuto komunikaci. OAuth 2.0 umožňuje obdobnou autentizaci jako na Obr. 4.3. Umožňuje i jiné dialogy - na obrázku je příklad dialogu protokolu OAuth 2.0, kdy je rozdělen dialog do dvou fází:

1. Autentizace, jejíž výsledkem je získání Autorizačního kódu, který může být náhodný, a tak nezadat šanci útočníkovi útočícímu na uživatelův počítač zneužít lístek zaslaný Autorizačním serverem.
2. Autorizace, kdy Klient (aplikace) získá přístupový lístek s oprávněními poskytnout uživateli příslušnou službu. Lístek se zde nazývá *Access Token*. Klient může získat i tzv. *Refresh Token*, sloužící k obnovení lístku.

4.4 RBAC model

Role-Based Access Control (RBAC) model [25] předpokládá, že subjekt má v rámci nějaké oblasti/domény/říše (např. organizace) jednu nebo více identit (Obr. 4.5). Každá jeho identita má konkrétní atributy (kontaktní údaje, hesla, certifikáty veřejných klíčů atp.). Oprávnění je možné dále seskupovat do rolí, které se dále mohou vkládat jedna do druhé. Role mohou být přiřazovány přímo uživatelům – mluvíme o základním RBAC modelu nebo jej můžeme rozšířit o tzv. pozice, role pak mohou být vztaženy i k pozici – rozšířený RBAC model.

Důležité ale je, že konkrétní oprávnění pro přístup a práci s aktivy nejsou přímo vázaná na identitu, ale na role. Tj. identitám jsou přiřazeny role a teprve na role jsou navázána oprávnění. Při změně role, tak automaticky dojde ke změně oprávnění. Roli tady můžeme svázat buď přímo s uživatelem, nebo, což je výhodnější, s pozicí v organizaci (většinou v praxi jedné pozicí odpovídá více rolí). Role může být ale třeba občan při styku občana se státní mocí.



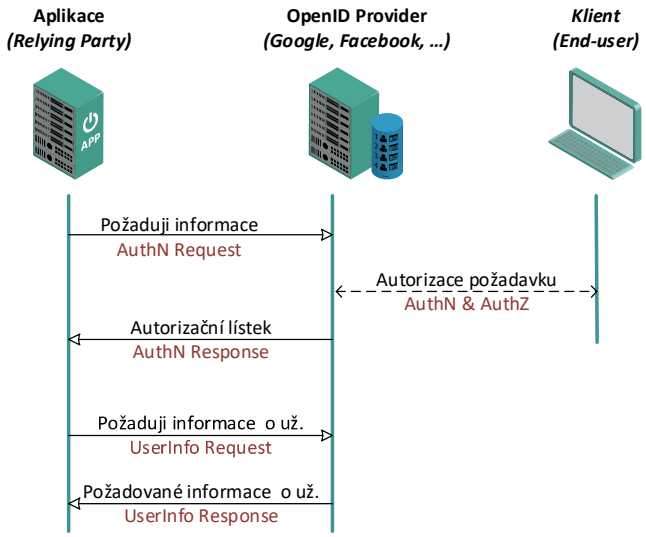
Obr. 4.5 RBAC model

4.4.1 OpenID Connect

Informace o uživateli udržuje zpravidla ověřovatel. V případě, že využíváme federaci identit, pak je užitečné získat atributy ověřené identity od prvotního ověřovatele. OpenID Connect [26] je protokol, který umožňuje získání atributů identity od původního ověřovatele.

Příklad (Obr. 4.6): Pro ověření do aplikace budeme využívat přihlášení do systému Facebook (v systému Facebook máme odkaz do uvedené aplikace). V případě, že uživatel přejde na tento odkaz, díky federaci identit se akceptuje autentizace ze systému Faceebook do

uvedené aplikace. Pro založení uživatele v aplikaci potřebujeme jeho atributy. Ty získáme přes protokol OpenID Connect.



Obr. 4.6 OpenID Connect

4.5 Autorizace

Autentizace ověřila identitu subjektu. Nyní identita chce přistupovat ke konkrétním zdrojům (aktivům). Proces, který přiřadí práva autentizované identitě pro přístup ke zdrojům, se nazývá autorizace.

Z Obr. 4.5 jakoby plynulo, že identita automaticky po autentizaci získá oprávnění sobě přiřazených rolí. Obecně tomu tak ale není. Proces autorizace totiž může být závislý na kvalitě (síle) použité autentizace. Identitě jsou pak přiřazeny jen ty role (z možných rolí, které mu mohou být přiřazeny), které odpovídají síle jim použité autentizační metody.

4.6 *Risk based* autorizace

Risk based autorizace je metodou hodnocení autentizačních mechanismů. Cílem Risk based autorizace, je pokud možno, automatizovaně (On Line) autorizovat subjekt na základě jím použitých autentizačních metod.

Problémem je, jak kvantifikovat kvalitu použité autentizace. Risk based autorizace využívá postupy obdobné analýze rizik. Tj. vychází se z empiricky stanovené míry rizika jednotlivých autorizačních metod. Tato metoda na první pohled nevypadá příliš exaktně, ale během následného vyhodnocování bezpečnostních incidentů („zpětná vazba“) mohu být rizika upravována, takže po určitém čase může tato metoda být velice efektivní.

4.6.1 Kategorie „Něco ví“

Pro kategorii autentizace „něco ví“ je praktické vycházet z tabulky bezpečnostních vlastností (např. Tabulka 1), která obsahuje n bezpečnostních vlastností (řádků). Vytvoříme obdobnou tabulku (Tabulka 4). Tuto tabulku doplníme o sloupec váhy, která vyjadřuje váhu rizika v_i i-té bezpečnostní vlastnosti. Přitom, aby při rozšíření tabulky o další vlastnosti se hodnocení zásadně nezměnilo, tak je vhodné, aby bylo splněno:

$$\sum_{i=1}^{n} v_i = 1$$

Pro k-té schéma pak v tabulce uvádíme, zda je bezpečnostní vlastnost r_i^k rizikem (=1) nebo není rizikem (=0). Kvalitu q^k pak můžeme vyjádřit jako:

$$q^k = 1 - \sum_{i=1}^{n} v_i \, r_i^k$$

Tabulka 4 Příklad hodnocení rizik pro kategorii "něco ví"

i	Bezpečnostní vlastnost	Váha (v_i)	Je rizikem (r_i^k)			
			Heslo (slabá autentizace)	*Secure Hash-Based Password Authentication Protocol Using Smartcards* [13]	*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy (kap. 1)*	AKA
1	Oboustranná autentizace	$^1/_{15}$	1	0	0	0
2	Změnu hesla součástí schématu	$^1/_{15}$	1	0	0	0
3	Nastavení hesla v případě zapomenutí hesla	$^1/_{15}$	1	0	0	0
4	Vylákání hesla	$^1/_{15}$	1	0	0	0
5	Odposlechnutí hesla	$^1/_{15}$	1	0	0	0
6	Uhodnutí hesla	$^1/_{15}$	1	0	0	0
7	Útoky na synchronizaci času	$^1/_{15}$	0	0	0	0
8	Útoky na synchronizaci komunikace	$^1/_{15}$	1	0	0	0
9	Odvolání datového nosiče	$^1/_{15}$	0	0	0	0
10	Anonymita autentizovaného subjektu	$^1/_{15}$	1	0	0	1
11	Nevystopovatelnost	$^1/_{15}$	1	0	0	1

12	*Session key agreement* [6]	$^1/_{15}$	1	1	0	0
13	*Perfect forward secrecy*	$^1/_{15}$	0	0	0	0
14	*Forward and backward secrecy*	$^1/_{15}$	0	0	0	0
15	*Key freshness*	$^1/_{15}$	0	0	0	0
q^k			$^5/_{15}$	$^{14}/_{15}$	$^{15}/_{15}$	$^{13}/_{15}$

4.6.2 Kategorie „Něco má“

Z bezpečnostních vlastností této kategorie lze udělat obdobnou tabulku jako je Tabulka 4. Jednotlivé bezpečnostní vlastnosti opět oceníme váhou w_i (Tabulka 5) za předpokladu:

$$\sum_{i=1}^{m} w_n = 1$$

Pro jednotlivá schémata vyjádříme riziko (0 nebo 1) hodnotou R_i^k. Výslednou kvalitu k-tého schématu získáme:

$$Q^k = 1 - \sum_{i=1}^{m} w_i \, R_i^k$$

Tabulka 5 Příklad hodnocení rizik pro kategorii "něco má"

i	Bezpečnostní vlastnost	Váha (w_i)	Je rizikem (R_i^k)			
			Heslo	*Secure Hash-Based Password Authentication Protocol Using Smartcards* [13]	*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy (kap. 1)*	*AKA*
1	Zařízení uchovává kryptografický materiál (a aplikace jej využívá).	$^1/_4$		0	0	0

[6] Hodnoty pro *Session key agreement, Perfect forward secrecy Forward and backward secrecy* a *Key freshness* nastavuji na 0 i pro metody, pro které tyto bezpečnostní vlastnosti nejsou relevantní.

2	Přístup ke kryptografickému materiálu pomocí hesla nebo PIN.	$^1/_4$		1	1	0
3	Kryptografický materiál neopouští zařízení (je neexportovatelný).	$^1/_4$		1	1	0
4	Zařízení je fyzicky chráněno proti neoprávněnému přístupu.	$^1/_4$		1	1	1
Q^k				$^1/_4$	$^1/_4$	$^3/_4$

4.6.3 Kategorie „Něčím je“

V této kategorii budeme zejména uvažovat faktor digitální stopa. Budeme vyhodnocovat korelaci mezi informacemi uloženými a aktuálně zjištěnými. Korelační koeficient ρ je z intervalu $< -1, 1 >$. Záporné hodnoty jsou důležité při zjištění abnormálních hodnot v digitální stopě. Pomocí nich je možné např. při zjištění některých abnormálních hodnot i snižovat celkovou váhu autorizace.

4.6.4 Federace identit

V případě přebírání autentizace od externího poskytovatele identit můžeme kvalitu empiricky stanovit v intervalu $(0, 1 >$.

4.6.5 Více faktorová autentizace

V případě více faktorové autentizace je výsledná kvalita autentizace součtem jednotlivých faktorů. Každý nový faktor je vážen nezávislostí (růzností) další autentizace na předchozích faktorech. V kapitole 0 je uvedeno, jak se mohou jednotlivé faktory lišit. Opět můžeme sestavit obdobnou tabulku (Tabulka 6).

Tabulka 6 Příklad vah vice faktorové autentizace

i	Liší se	Váha (W_i)	Je rizikem (R_i^k)			
			Heslo	*Secure Hash-Based Password Authentication Protocol Using Smartcards* [13]	*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy (kap. 1)*	AKA
1	Různým kryptografickým materiálem	$^1/_4$	0	0	0	0
2	Různým autentizačním schématem	$^1/_4$	0	0	0	0
3	Různým komunikačním protokolem	$^1/_4$	0	1	1	1

4	Různým komunikačním kanálem	$^1/_4$	0	1	1	1
Liší se			0	$^1/_2$	$^1/_2$	$^1/_2$

4.6.6 Příklad 1

Secure Hash-Based Password Authentication Protocol Using Smartcards [8] (kap. 4.1.1) je dvoufaktorová autentizace:

- „Něco ví“ (heslo) – byla ohodnocena váhou $^{15}/_{15}$ (Tabulka 4).
- „Něco má“ (datový nosič) – byla ohodnocena váhou $^1/_4$ (Tabulka 5). Jenže tento druhý faktor budeme vážit hodnotou $^1/_2$ podle Tabulka 6.

Výsledná kvalita je $^{15}/_{15} + ^1/_4 \cdot ^1/_2 = 9/8$

4.6.7 Příklad 2

Pro aplikaci elektronického bankovnictví bylo stanoveno, že subjekt se musí pro roli zjištění zůstatku na účtu autentizovat nejméně s kvalitou 0,7 a pro roli podání platebního příkazu nejméně s kvalitou 1.

Subjekt má k dispozici:

- Autentizaci heslem s kvalitou $^1/_3$ (Tabulka 4).
- Autentizaci pomocí externího (federativního) poskytovatele identit Facebook stanovenou na 0,5.
- *Secure Hash-Based Password Authentication Protocol Using Smartcards* stanovenou na $9/8$. (kap. 4.6.6).

V případě jedno faktorové autentizace může použít jen schéma *Secure Hash-Based Password Authentication Protocol Using Smartcards*.

V případě více faktorové autentizace může pro roli zůstatku na účtu použít i kombinaci autentizaci pomocí hesla a autentizaci pomocí poskytovatele Facebook. Protože

- První faktor je heslo s kvalitou $^1/_3$.
- Druhý faktor je ohodnocen na 0,5 ale liší se ve všech 4 bodech dle Tabulka 6.

Výsledná kvalita této dvou faktorové autentizace je $^5/_6$ a to je větší než 0,7.

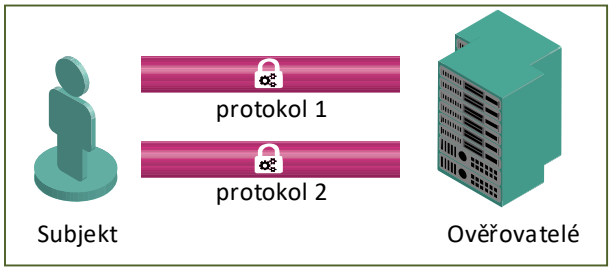
4.6.8 Příklad 3

Po čase provádíme vyhodnocení aplikace podle Příkladu 2. Zjistíme, že dochází k bezpečnostním incidentům. Předpokládaným důvodem bezpečnostních incidentů je federace s poskytovatelem Facebook. Na základě tohoto zjištění snížíme stanovenou kvalitu autentizace pomocí poskytovatele Facebook z 0,5 na 0,3. Důsledkem bude, že více faktorová

autentizace heslem a pomocí poskytovatele Facebook nebude již možná (např. po dobu šetření uvedených bezpečnostních incidentů).

5 Cíle dizertační práce

Více faktorová autentizace je efektivnější při různosti autentizačních faktorů. Přitom možnosti využívání více komunikačních kanálů k autentizaci je velice omezená[7]. Za přínosnou metodu považuji autentizaci za využití více ověřovatelů.



Obr. 5.1 Více faktorová autentizace s více ověřovateli

Cílem práce je navrhnout nové autentizační algoritmy, které využívají více nezávislých ověřovatelů. Přitom je důležité, aby se nejednalo o více jedno faktorových autentizací použitých za sebou, ale o více faktorovou autentizaci. Tj. aby autentizace byly provázány.

Navrhl jsem dva autentizační algoritmy využívající více nezávislých ověřovatelů. První z nich již byl publikován v [2] a [1], druhý je uveden v kapitole 1.

Cíle další práce jsou:

- Na základě bezpečnostní analýzy porovnat navržená řešení.
- Publikovat řešení navržené v kapitole 1, včetně bezpečnostní analýzy tohoto řešení.
- Na základě bezpečnostní analýzy protokolů používaných v mobilních sítích navrhnout implementaci autentizace využívající více nezávislých ověřovatelů.
- Navrhnout poskytovatele identit založeného na bázi navržených autentizačních protokolů (viz 4.3.1, 4.3.2).

Dalším možným směrem výzkumu je oblast využití a modelování autentizace na základě digitální stopy. Digitální stopu lze prakticky využít nejenom k samotné autentizaci subjektu, ale je zajímavé ji sledovat i během již navázané relace. Např. server může do běžící relace vkládat cílená metadata a testovat reakci klienta. Na základě této reakce může usuzovat na pravost subjektu.

[7] Např. využívání zasílání jednorázových hesel pomocí SMS bylo zajímavé až do okamžiku zavedení tzv. „chytrých telefonů“.

6 Publikace autora

6.1 Publikace bezprostředně se týkající tématu

- L. Dostálek: Authentication and authorization applications in 4G networks, Konference: Security and protection of information, Brno 2015, ISSN 2336-5587, ISBN 978-80-7231-997-8
- L. Dostálek, J. Ledvina: Strong Authentication for Internet Mobile Application, Konference: Applied Electronic, Plzeň 2015, IEEE CFP1569A-PRT, ISBN 978-80-261-0385-1, ISSN 1803-7232

6.2 Ostatní publikace

- Martin Dvorak, Libor Dostalek, Zora Rihova: Optimizing the Amount of Data to Evaluate the Events of Cyber Security, International Journal of Modern Communication Technologies & Research (IJMCTR), ISSN: 2321-0850, Vol. 3 Issue 7 (July 2015)
- L. Dostálek, M. Novák: The Cryptographic Sensor, konference: Security and Protection of Information, Brno 2013
- L. Dostálek, I. Dostálková: Není PDF/A jako PDF/A, Data Security Management 1/2013: ISSN 1211-8737
- L. Dostálek: Formáty pro zaručené elektronické podpisy – část IV, Data Security Management 3/2012: ISSN 1211-8737
- L. Dostálek, K. Štíchová: Biometrický podpis v PDF – Formát PDF, Data Security Management 2/2012: ISSN 1211-8737
- L. Dostálek: Formáty pro zaručené elektronické podpisy – Formát PDF, Data Security Management 1/2012: ISSN 1211-8737
- L. Dostálek: Formáty pro zaručené elektronické podpisy – Viditelný podpis a podpis ve formátu PDF, Data Security Management 4/2011: ISSN 1211-8737
- L. Dostálek: Formáty pro zaručené elektronické podpisy – část II,CAdES, Data Security Management 3/2011: ISSN 1211-8737
- L. Dostálek: Formáty pro zaručené elektronické podpisy – část I, Data Security Management 2/2011: XAdES, ISSN 1211-8737
- L. Dostálek, M. Vohnoutová: Velký průvodce infrastrukturou PKI a technologií elektronického podpisu, 534 stran, Computer Press, Druhé vydání 2010.

- L. Dostálek, I. Dostálaková: Password Audit as indicator of security qualaity, Systémová integracee 3/2009, VŠE Praha 2009, ISSN 1210-9479
- L.Dostálek, A.Kabelová: Velký průvodce TCP/IP a systémem DNS, 418 stran, Computer Press 1999, druhé vydání 2000, třetí vydání 2003, čtvrté vydání 2005 už 542 stran, páté vydání 2008.
- L.Dostálek, A.Kabelová: TCP/IP и DNS в теории и на практике. Полное руководство. Наука и техника 2006. ISBN 5-94387-280-9, Rusko
- L.Dostálek: Bezpieczeństwo protokołu TCP/IP, 768 stran, Wydawnictwo Naukowe 2006, ISBN 1083-01-14959-0, Polsko.
- L.Dostálek, A.Kabelová: Understanding TCP/IP, 462 stran, ISBN 1-904811-71-X, Packt Publishing 2006.
- L.Dostálek, A.Kabelová: DNS in Action, 183 stran, Packt Publishing 2006.
- J.M.Kretchmar, L.Dostálek: Administrace a diagnostika sítí, Computer Press 2004
- M.Vohnoutová, L. Dostálek, J.Lapáček: Připojme se k Internetu, Computer Press 2003.

- I. Dostalkova, L. Dostalek: The Impact of Synchronization on the Group Size, ICNAAM 2011, AIP Volume 1389, 2011 Halkidiki, GREECE

7 Dodatek

7.1 Robustní dvou-faktorová autentizace

Toto autentizační schéma Robustní dvoufaktorová autentizace (*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy* [9]) se skládá z pěti fází: generace parametrů: registrace, autentizace, změna hesla a odvolání datového nosiče.

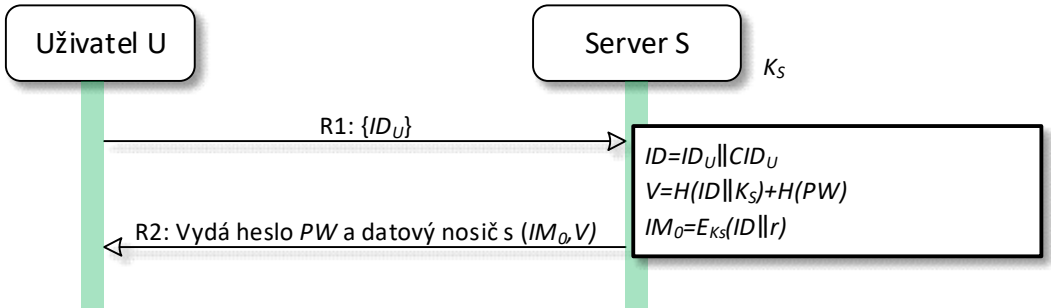
p	Velké přirozené číslo
E	Eliptická křivka nad tělesem Fp
F_p	Končené těleso, ve kterém budeme označovat: sčítání (+), odčítání (-) a násobení (x)
G	Generující bod velkého řádu
S	Server
U	Uživatel
ID	Identita uživatele U
PW	Heslo U
$E_{key}(m)$	Šifrování zprávy m klíčem key
$D_{key}(m)$	Dešifrování zprávy m klíčem key.
$h_1()$, $h_2()$, $h_3()$	Hešovací funkce
$H()$	Jednocestná funkce, jejímž výsledkem je bod na eliptické křivce (*map-to-point on elliptic curve hash function*) – viz např. [28]
$\oplus$	Operace XOR
$\|\|$	Operace sřetězení

7.1.1 Generace parametrů

S zvolí eliptickou křivku E nad tělesem F_p, kde p je velké prvočíslo. S dále zvolí generující bod G velkého řádu n. Nakonec S zveřejní parametry (p, E, G, n). S si udržuje své tajemství K_s.

7.1.2 Registrace

V této fázi se U registruje u S v následujících krocích (Obr. 7.1):



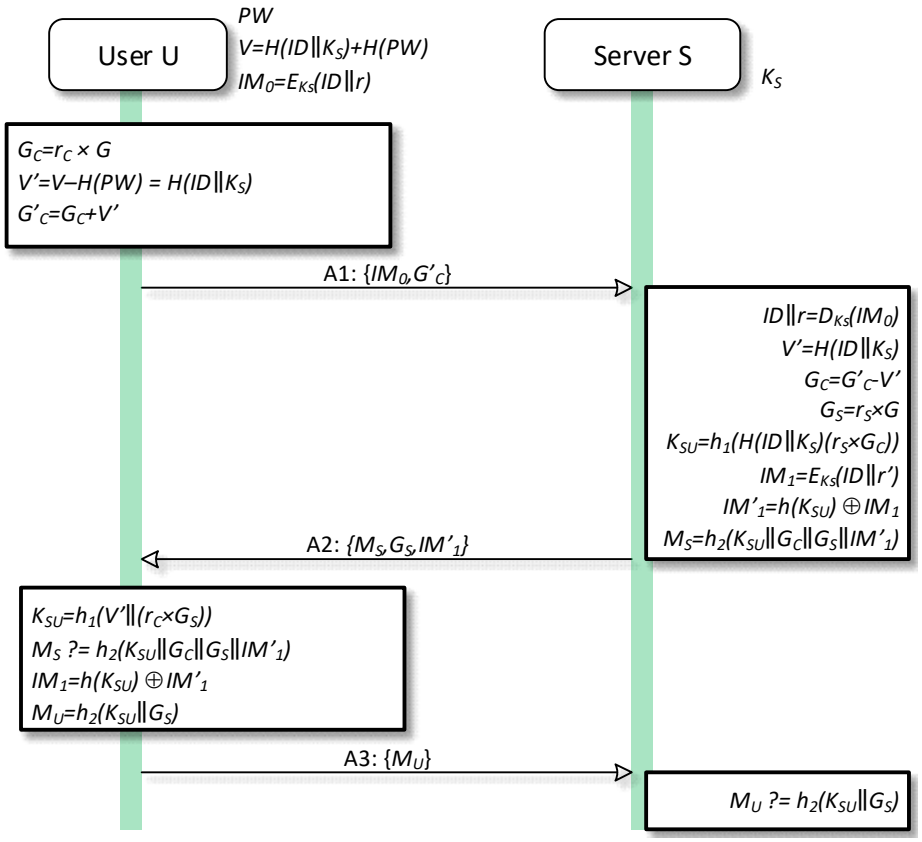
Obr. 7.1 Registrace

Krok R1. *U* si zvolí identifikátor ID_U a zašle jej *S*.

Krok R2. *S* přijme ID_U a vygeneruje identifikátor $ID=ID_U\|CID_U$ pro *U*. Kde CID_U je identifikátor datového nosiče *U*. Nyní *S* spočte $V=H(ID\|K_S)+H(PW)$ a $IM_0=E_{K_S}(ID\|r)$, kde K_S je hlavní tajemství, *PW* inicializační heslo zvolené *S* a *r* je náhodné číslo, které bude sloužit k ochraně identity.

Krok R3. *S* vydá heslo *PW* a datový nosič uživateli *U*, na kterém je uloženo (IM_0,V).

7.1.3 Autentizace



Obr. 7.2 Autentizace

V této fázi se vzájemně autentizují *U* a *S*. Dále stanoví klíč relace K_{SU}, kterým se bude zabezpečovat následná komunikace (Obr. 7.2):

Krok A1. *U* vloží datový nosič do zařízení a zadá heslo *PW*. Následně se vygeneruje náhodné číslo r_C z intervalu $[1, n-1]$ a spočte $G_C=r_C \times G$. Dále spočte $V'=V–H(PW) = H(ID\|K_S)$ a $G'_C=G_C+V'$. Nakonec *U* odešle dvojici $\{IM_0, G'_C\}$ serveru *S*.

Krok A2. *S* příjme $\{IM_0,G'_C\}$, dešifruje parametr IM_0 klíčem (tajemstvím) K_S a obdrží hodnotu $ID\|r$. Nyní *S* ověří, zda je identifikátor *ID* platný. Když ne, ukončí komunikaci. V případě, že identifikátor je platný, pak *S* spočítá $V'=H(ID\|K_S)$ a obnoví $G_C=G'_C-V'$. Pak *S* vygeneruje $G_S=r_S\times G$, kde r_S je náhodné celé číslo z intervalu $[1, n-1]$. Dále spočítá $IM_1=E_{K_S}(ID\|r')$, $K_{SU}=h_1(H(ID\|K_S)(r_S\times G_C))$, $IM'_1=h(K_{SU}) \oplus IM_1$ a $M_S=h_2(K_{SU}\|G_C\|G_S\|IM'_1)$. Trojici $\{M_S,G_S,IM'_1\}$ odešle *U*.

Krok A3. U přijme trojici $\{M_S,G_S,IM'_1\}$ a spočítá klíč relace $K_{SU}=h_1(V'\|(r_C\times G_S))$. Dále ověří, zda se hodnota M_S rovná $h_2(K_{SU}\|G_C\|G_S\|IM'_1)$. Jestliže ne, pak ukončí relaci. Jinak spočítáte $IM_1=h(K_{SU}) \oplus IM'_1$ a zamění IM_0 novou hodnotou IM_1. Nakonec spočítáte $M_U=h_2(K_{SU}\|G_S)$ a odešle *S*.

Krok A4. S příjme $\{M_U\}$ a zkontroluje, zda hodnota M_U se rovná $h_2(K_{SU}\|G_S)$. Když ano, *U* a *S* jsou vzájemně autentizování a mohou začít komunikovat. Komunikaci si zabezpečí sdíleným klíčem relace K_{SU}.

7.1.4 Změna hesla

U si v této fázi mění heslo *PW* na *PW**:

Krok PW1. *U* vloží do mobilního zařízení datový nosič a zadá své staré heslo *PW* a posečká na výzvu o zadání nového hesla *PW**.

Krok PW2. Mobilní zařízení spočítá: $V^* = V - H(PW) + H(PW^*)$, a zamění *V* za *V**.

7.1.5 Odvolání datového nosiče

Tato fáze umožní uživateli používat původní identitu, i když byl odvolán jeho datový nosič. *S* vygeneruje uživateli *U* identifikátor $ID_{new} = ID_U\|CID_{Unew}$, který je identifikátorem jeho nového datového nosiče. Nyní S spočítá $V_{new} = H(ID_{new}\|K_s) + H(PW)$ a $IM0_{new} = E_{K_S}(ID_{new}\|r)$. Server nyní vydá U nový datový nosič obsahující ($IM0_{new}$, V_{new}) a ve své tabulce si zamění *ID* za ID_{new}.

7.2 Navržená nová autentizační metoda

V tomto řešení navrhuji využít společně AKA schéma [3] a Robustní dvoufaktorovou autentizaci (*Robust Two-Factor Authentication and Key Agreement Preserving User Privacy* [9]) popsanou v kap. 1.

Předpokládáme, že uživatel je registrován tj.:

- V terminologii [9] to znamená, že si uživatel a server vyměnili registrační zprávy R1 a R2 (Obr. 7.1).
- V terminologii AKA schéma [3] to znamená, že uživatel je vybaven např. ISIM, který sdílí sdílené tajemství K s HSS.

Autentizace v tomto řešení proběhne v následujících krocích (Obr. 7.3):

Krok Y 1. *U* vloží do mobilního zařízení datový nosič, zadá heslo *PW*. Mobilní zařízení nejprve vygeneruje náhodné číslo r_C z intervalu $[1, n-1]$ a spočítá $G_C = r_C \times G$. Dále spočítá $V' = V - H(PW) = H(ID\|K_S)$ a $G'_C = G_C + V'$. Nakonec *U* odešle svou identitou účastníka (pro AKA schéma) i dvojici $\{IM_0, G'_C\}$ serveru *S*.

Krok Y 2. Tento krok obsahuje v sobě kroky A1 (kap. 7.1.3), AKA1, AKA2 a AKA3 [3].

S přijme zprávu Y1. Následně požádá HSS (resp. AuC) o generování autentizačního vektoru *AV* pro účastníka. HSS vygeneruje autentizační vektor *AV*. *S* z *AV* vyzobne a uloží *XRES*.

Mezitím *S* zpracovává přijaté $\{IM_0, G'_C\}$. Dešifruje parametr IM_0 klíčem (tajemstvím) K_S a obdrží hodnotu $ID\|r$. Nyní *S* ověří, zda identifikátor *ID* je platný, když ne, ukončí komunikaci. V případě, že identifikátor je platný, pak *S* spočítá $V' = H(ID\|K_S)$ a obnoví $G_C = G'_C - V'$. Poté *S* vygeneruje $G_S = r_S \times G$, kde r_S je náhodné celé číslo z

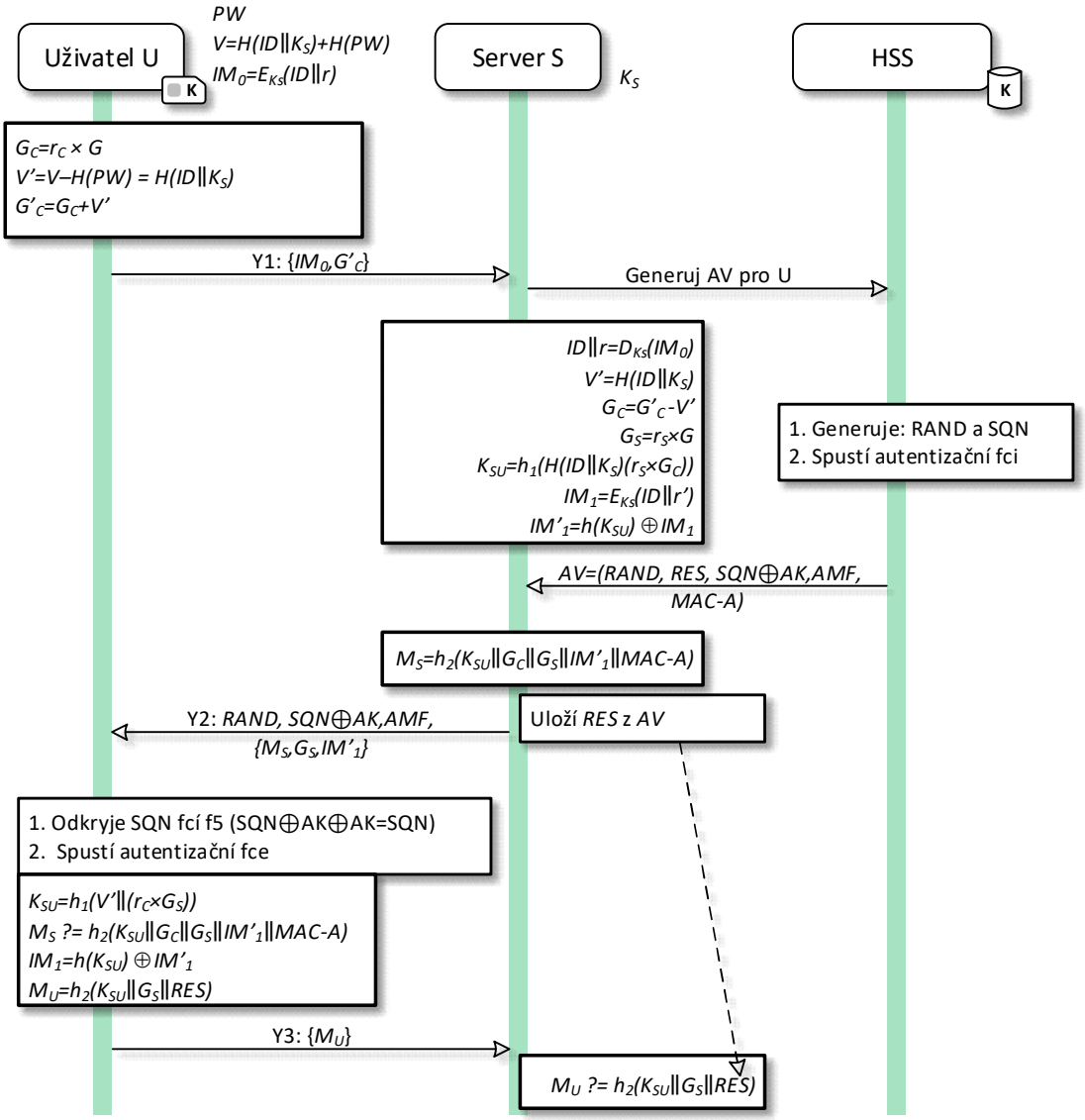
intervalu $[1, n-1]$. Dále spočítá $IM_1=E_{Ks}(ID\|r')$, $K_{SU}=h_1(H(ID\|K_S)(r_S\times G_C))$, $IM'_1=h(K_{SU}) \oplus IM_1$ a $M_S=h_2(K_{SU}\|G_C\|G_S\|IM'_1\|MAC\text{-}A)$. T

S odešle *U* jednak trojici $\{M_S,G_S,IM'_1\}$ a jednak $RAND, SQN\oplus AK,AMF$.

Krok Y 3. Zařízení uživatele *U* nejprve spustí funkci f5, aby získalo *SEQ*, které porovná s jim udržovaným *SEQ*. Poté spustí zbylé jednocestné funkce a získá *MAC-A, RES* a kryptografický materiál *IK, CK*.

Z přijaté trojice $\{M_S,G_S,IM'_1\}$ spočítá klíč relace $K_{SU}=h_1(V'\|(r_C\times G_S))$. Dále ověří, zda je přijatá hodnota M_S se rovná $h_2(K_{SU}\|G_C\|G_S\|IM'_1\|MAC\text{-}A)$. Jestliže ne, pak ukončí relaci. Jinak spočítá $IM_1=h(K_{SU}) \oplus IM'_1$ a zamění IM_0 novou hodnotou IM_1. Nakonec spočítá $M_U=h_2(K_{SU}\|G_S\|RES)$ a odešle *S*.

Krok Y 4. *S* příjme $\{M_U\}$ a zkontroluje, zda hodnota M_U se rovná $h_2(K_{SU}\|G_S\|XRES)$. Když ano, *U* a *S* jsou vzájemně autentizování a mohou začít komunikovat. Komunikaci si zabezpečí sdíleným klíčem relace K_{SU}.



Obr. 7.3 Navržené řešení

8 Citovaná literatura

[1] L. Dostalek, „Authentication and authorization applications in 4G networks," v *Security and protection of information, ISSN 2336-5587, ISBN 978-80-7231-997-8*, Brno 2015, 2015.

[2] L. Dostalek a J. Ledvina, „Strong Authentication for Mobile Application," *International Conference ob Applied Elecronics,* č. IEEE CFP1569A-PRT, pp. 23-26, September 2015.

[3] „3G security; Security architecture," 3GPP TS 33.102 , December 2014. [Online]. Available: http://www.3gpp.org.

[4] „3GPP System Architecture Evolution (SAE); Security architecture," 3GPP TS 33.401, December 2014. [Online]. Available: http://www.3gpp.org/.

[5] „3G security; Access security for IP-based services," 3GPP TS 33.203, December 2014. [Online]. Available: http://www.3gpp.org.

[6] A. Niemi, J. Arkko a V. Torvinen, „ Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)," IETF RFC 3310, September 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3310.txt.

[7] W. C. Ku, „A hash-based strong-password authentication, scheme without using smart card," *ACM Operating Systems Review, 38(1),* p. 29–34, 2004.

[8] H. Jung, H. S. Kim, B. Murgante, O. Gervasi a A. Iglesias, „Secure Hash-Based Password Authentication Protocol Using Smartcards," v *11th International Conference on Computational Science and Its Applications (ICCSA), PT V Book Series: Lecture Notes in Computer Science, Volume: 6786, Pages: 593-606*, 2011.

[9] Q. Jiang, J. Ma, G. Li a L. Yang, „ Robust Two-Factor Authentication and Key Agreement Preserving User Privacy," *IJ Network Security, 16(4),* pp. 321-332, 2014.

[10] „IP Multimedia Subsystem (IMS); Stage 2," 3GPP TS 23.228, September 2015. [Online]. Available: http://www.3gpp.org.

[11] H. Jeong, D. Won a S. Kim, „Weaknesses and improvement of secure hash-based strong password authentication protocol," *Journal of Information Science and Engineering,* sv. 26, p. 1845–1858, 2010.

[12] L. Lamport, „Password Authentication with Insecure Communication," *Communications of the ACM,* sv. 24, č. 11, pp. 770-772, 1981.

[13] M. Kim a C. K. Koc, „A secure hash-based strong-password authentication protocol using one-time public-key cryptography," *Journal of Computer and Systems Sciences International,* č. 45, p. 623–626, 2006.

[14] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, J. W. Zhang a Z. Y. Feng, „Improvements of Juang's password authenticated key agreement scheme using smart cards," *IEEE Transactions on Industrial Electronics,* sv. 56, č. 6, pp. 2284-2291, 2009.

[15] „Secure Element Access Control," GlobalPlatform Device Technology, 2012. [Online]. Available: http://www.globalplatform.org.

[16] „TEE Protection Profile, Version 1.0," GlobalPlatform Device Committee, 2013. [Online]. Available: http://www.globalplatform.org/.

[17] L. Adamec, „Testování biometrického systému založeného na dynamice podpisu," *Masarykova Univerzita Brno, Diplomová práce,* 2011.

[18] R. M. Needham a M. D. Schroeder, „Using encryption for authentication in large networks of computers," *Communications of the ACM,* pp. 993-999, Volume 21 Issue 12, Dec. 1978 .

[19] C. Neuman, T. Yu, S. Harman a K. Raeburn, „The Kerberos Network Authentication Service (V5)," IETF RFC 4120, July 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4120.txt.

[20] „Assertions and Protocols for the OASIS, Security Assertion Markup Language," OASIS, 15 March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/.

[21] D. Hardt, „The OAuth 2.0 Authorization Framework," IETF RFC 6749, October 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6749.txt.

[22] M. Jones a D. Hardt, „The OAuth 2.0 Authorization Framework: Bearer Token Usage," IETF RFC 6750, October 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6750.txt.

[23] J. Richer, „ OAuth 2.0 Token Introspection," IETF RFC 7662, October 2015. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7662.txt.

[24] M. Jones, J. Bradley a N. Sakimura, „JSON Web Token (JWT)," IETF RFC 7519, May 2015. [Online]. Available: JSON Web Token (JWT).

[25] D. F. Ferraiolo a D. R. Kuhn, „Role-Based Access Controls," v *15th National Computer Security Conference*, Baltimore MD, 1992.

[26] D. Hardt, „The OAuth 2.0 Authorization Framework," IETF RFC 6749, October 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6749.txt.

[27] M. Jones a D. Hardt, „The OAuth 2.0 Authorization Framework: Bearer Token Usage," IETF RFC 6750, [Online]. Available: http://www.rfc-editor.org/rfc/rfc6750.txt.

[28] T. Icart, „How to hash into elliptic curves," v *CRYPTO 2009*, Santa Barbara, California, USA, 2009.

[29] D. Eastlake a T. Hansen, „US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)," IETF RFC 6234, May 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6234.txt.

[30] „3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General," 3GPP TS 35.205, September 2014. [Online]. Available: http://www.3gpp.org.

[31] „3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm specification," 3GPP TS 35.206, September 2014. [Online]. Available: http://www.3gpp.org.

[32] W.-C. Ku a J. Fu, „A hash-based strong-password authentication scheme without using smart cards," *ACM SIGOPS Operating Systems Review ,* sv. 38, č. 1, pp. 29-34, 2004.

[33] „OpenID Specifications," OpenID Foundation, 2015. [Online]. Available: http://openid.net/developers/specs/.