

Kryptologie pro praxi – autentizace

Tématem tohoto dílu je využití kryptografických metod k bezpečnému prokázání identity subjektů v informačních systémech. Nejprve si ozřejmíme rozdíl mezi pojmy identifikace a autentizace, které jsou někdy v teoretických pramenech považovány za synonyma. Z praktického hlediska označujeme slovem identifikace prosté přiřazení určité, byť jen domnělé identity, zatímco výraz autentizace subjektu znamená její bezpečné potvrzení. Identifikace tak probíhá třeba na základě prohlášení: „Já jsem Frank Drebin!“, což může udělat prakticky kdokoliv a význam tohoto kroku je čistě inženýrský – ověřovací aplikace vyhledá složku „Frank Drebin“ a rozhodne, jakým způsobem bude provádět autentizaci. Teprve po úspěšném projití autentizační fázi je osoba sedící u terminálu vpuštěna do systému pod identitou Lt. Frank Drebin, LA Police. Občas se ještě zavádí následná fáze autorizace, ve které se rozhoduje o přidělení sady oprávnění pro práci v systému, což je opět většinou fáze čistě inženýrská. Naše pozornost bude dále soustředěna na fázi autentizace subjektu. Zdůrazněme zde, že kryptologie striktně rozlišuje dva základní druhy autentizace a to autentizaci subjektu a autentizaci původu zprávy. Rozdíl mezi nimi je v tom, vzhledem k čemu se prokazovaná identita vztahuje. První druh slouží k prokázání identity vzhledem k nějakému bodu časoprostoru (tady a teď sedí poručík Drebin). Druhá skupina metod pak prokazuje identitu vzhledem k nějakému dokumentu (tento příkaz poslal poručík Drebin) a v praxi se většinou realizuje pomocí schémat digitálního podpisu (viz předchozí díly seriálu). Za jistých okolností lze schémata z obou kategorií navzájem převádět. Musíme ovšem velmi pečlivě hlídat, jestli se bezpečnostní vlastnosti převáděného schématu správně transformují do cílového modelu.

Základní schémata

Níže uvedená schémata lze chápat jako elementární stavební prvky každého autentizačního protokolu. Jejich cílem je propojení úlohy prolomení autentizace s jinou úlohou, která je obecně v kryptologii považována za neschůdnou. Pro jednoduchost se zaměříme na jednosměrnou autentizaci uživatele (U) do nějakého systému (S). Relaci zaslání zprávy mezi uživatelem a systémem budeme zapisovat obvyklou symbolikou: zdroj -> cíl: zpráva. Elementárním bezpečnostním požadavkem je, aby se útoč-

ník nemohl do systému přihlásit na základě odposlechu vyměňovaných zpráv. Tím hned v úvodu padá možnost přihlašování se prostým zasláním uživatelského hesla. Obecný scénář všech níže uvedených schémat lze popsat ve třech krocích:

- 1) $U \rightarrow S: id_U$,
- 2) $S \rightarrow U: výzva$,
- 3) $U \rightarrow S: odpověď$.

Na základě identifikátoru uživatele id_U zasláního v prvním kroku vyhledá systém přihlašovací informace a rozhodne mj. jakým způsobem sestaví výzvu pro druhý krok. Na tuto výzvu uživatel reaguje zadáním příslušné odpovědi, kterou zašle systému k ověření. Pokud ověření dopadne kladně, je uživateli přiřazena identita id_U , v opačném případě protokol končí chybovým stavem. Dále si postupně ukážeme čtyři základní způsoby, kterými lze sestavení a vyhodnocení posloupnosti výzva-odpověď realizovat. Cílem je zajistit, aby k náhodně volené výzvě dokázal sestavit správnou odpověď právě jen oprávněný majitel identity id_U . Dodejme, že současnou oblast autentizačních schémat lze na základní a odvozené metody rozdělit mnoha různými způsoby. Náš výklad se kromě vlastních zkušeností opírá také o uznávaný zdroj [2].

Jako první možnost se nabízí symetrická šifra. V takovém případě je výzva generována jako náhodné číslo z dostatečně velkého intervalu. Reakce uživatele se počítá jako $odpověď = E_K(výzva \parallel id_S)$, kde id_S je jednoznačný identifikátor systému, do kterého se uživatel přihlašuje. Operace $E_K(m)$ zde znamená šifrování zprávy m způsobem, který zároveň zaručuje důvěrnost i integritu (viz ST 9/2003 a ST 2/2004). Klíč K zde představuje sdílené tajemství mezi uživatelem a systémem. Zpracování odpovědi na straně systému spočívá v jejím odšifrování (včetně ověření integrity) a kontrole obsažených položek oproti jejich vzorovým hodnotám. Úkolem identifikátoru id_S , který se bude objevovat i v ostatních schématech, je komplikovat aktivní útoky založené na přeměňování spojení. Mělo by zde být obsaženo maximum informací, které uživatel během autentizace o systému ví (včetně IP adresy terminálu a serveru, atp.). Z bezpečnostního hlediska má být id_S při kontrole na straně serveru odvozen stejně jako na straně uživatele z kontextu navázaného spojení.

Další možností je využití hašovací funkce. Generování výzvy probíhá stejně

jako v předchozím případě. Reakce uživatele je počítána podle vzorce $odpověď = h_K(výzva \parallel id_S)$, kde h_K je klíčovaná hašovací funkce, například podle schématu HMAC (ST 2/2004). Význam ostatních parametrů je vysvětlen výše. Ověření na straně serveru probíhá výpočtem kontrolní odpovědi a jejím srovnáním s odpovědí zaslouanou uživatelem. Uživateli je přiřazena jím proklamovaná identita právě tehdy, když jsou obě odpovědi shodné.

Při autentizaci se může uplatnit i asymetrická šifra. Pro předcházení útokům s voleným šifrovým textem a přeměňováním spojení se nyní postupuje odlišně už při generování výzvy. Systém sice opět nejprve vygeneruje náhodné číslo (r), avšak dále se vypočítá $výzva = [h(r), E_U(r \parallel id_S)]$, kde h je jednosměrná hašovací funkce a $E_U(m)$ je asymetrické šifrování zprávy m veřejným klíčem uživatele. Vhodným schématem je zde například RSA s kódováním OAEP (ST 10/2003 a 3/2004). Uživatel výzvu nejprve pomocí privátního klíče odšifruje, čímž získá hodnoty r a id_S . Potom ověří identifikátor systému (jeho IP adresu, atp.) a zkontroluje platnost vztahu $výzva = h(r)$. Pokud vše souhlasí, zašle systému $odpověď = r$. Použití asymetrické šifry eliminuje určitou nevýhodu předchozích schémat, kterou je nutnost bezpečného sdílení a hlavně dohodnutí tajného klíče K mezi systémem a uživatelem. Cenou je zde podstatně vyšší výpočetní náročnost. Toto schéma se také snadno propojuje s generováním klíčů pro zajištění autentizovaného kanálu (viz dále).

Jako nejrobustnější základ můžeme nasadit podpisové schéma. V takovém případě je výzva generována opět jako náhodné číslo, na které uživatel zasílá $odpověď = [r_A, S_U(r_A \parallel výzva \parallel id_S)]$, čili svůj digitální podpis zprávy složené z náhodného čísla generovaného uživatelem, výzvy a identifikátoru systému. Úkolem r_A je bránit útokům na podpisové schéma ze strany kompromitovaného systému. Na základě certifikátu veřejného klíče uživatele, který systém vyhledal během identifikační fáze protokolu (v některých případech o něj může uživatel přímo požádat), ověří pravost podpisu a v případě kladného výsledku vpustí uživatele do systému. Toto schéma dále vylepšuje předchozí metody směrem k nepopíratelnosti autentizace, kdy uživatel nemůže jednoduše popřít, že se v daný čas z daného místa do systému přihlásil. V ostatních případech má totiž systém k dispozici jako důkaz jen

takové hodnoty, které si dokáže sám vytvořit, takže uživatel se může například před soudem bránit, že celá autentizace byla fingována s cílem „příšit“ mu nějaký čin, který nespáchal. Autentizace založená na výše popsaných schématech může být také nepopíratelná, avšak je nutné, aby do protokolu interaktivně vstupovala nějaká důvěryhodná třetí strana.

Specifika praxe

Obecně můžeme všechny principy autentizace subjektu rozdělit do tří elementárních kategorií, které se často navzájem kombinují: autentizace tím, co subjekt *zná* (heslo, PIN), tím, co subjekt *má* (tzv. předmět – čipová karta, dnes zejména SIM pro GSM), a konečně tím, co subjekt *je* (biometrické údaje – otisk palce, duhovka, hlas, atp.). Kryptologie se zabývá zejména první a druhou třídou, přičemž z praktického hlediska můžeme uvést, že se zde vždy setkáme s jistou modifikací výše uvedených základních schémat. Tyto modifikace jsou ovšem často podstatné, což odráží pochopitelnou variabilitu podmínek, ve kterých má být konkrétní autentizace prováděna. Nejméně úprav si vyžadují metody z druhé kategorie, kdy se do předmětu, jehož vlastnictvím se má uživatel prokázat, jednoduše uloží privátní klíč nebo sdílený symetrický klíč.

Více úprav si ovšem žádá první druh schémat a to díky principiálnímu problému, kterým je nízká entropie hesel a PINů, které je průměrný uživatel schopen si zapamatovat. Pokud bychom například ve výše uvedených schématech místo symetrického klíče použili rovnou derivát uživatelského hesla, zjistíme, že výsledný protokol je za elementárního předpokladu odpovídajícího útočníka snadno zranitelný pomocí slovníkového útoku. V praxi najdeme dva základní přístupy k řešení tohoto problému: Prvním je kombinace hesla a předmětu, kdy se uživatel nejprve lokálně autentizuje vůči svému předmětu, kterým se potom autentizuje do vlastního systému. Tento model odpovídá nejen mobilním bankovním aplikacím, ale například i situaci, kdy se uživatel autentizuje

privátním klíčem, který má šifrovaný svým heslem. Slovníkový útok tu připadá v úvahu jen při primární, lokální autentizaci, kde ho musíme eliminovat technickými prostředky. Druhým způsobem je použití zvláštního protokolu, který slovníkovému útoku brání komplikovaným „propletením“ jednosměrných funkcí a náhodných čísel. Tyto protokoly jsou velmi specifické a často musí být postaveny na míru podle konkrétního modelu hrozeb a možností koncových zařízení. Dobrou zprávou pro architektu systémů nicméně je, že vůbec existují. Jejich nejpočetnější rodina se vzhledem k implicitnímu propojení s autentizovaným kanálem (viz níže) obvykle označuje jako Password-Authenticated Key Exchange. Jedním z posledních přírůstků do této rodiny jsou protokoly EPA/EPA+ [1].

Upozorníme, že autentizace subjektu sama o sobě ještě nezajišťuje integritu přenášených zpráv. V praxi se lze setkat s aplikacemi, které na začátku uživatele sice pečlivě autentizují, avšak dál už se nestarají o to, jestli jeho terminál někdo neodpojil a nepřevzal spojení místo něho. V bezpečných aplikacích je nutné autentizaci a integritu (někdy i důvěrnost) propojit a tím vytvořit autentizovaný kanál: na začátku sezení má proběhnout bezpečné potvrzení identity, jejímž výsledkem je mj. i odvození klíčů pro ustanovení chráněného spojení, které zajišťuje důvěrnost a integritu přenášených zpráv. Příkladem takového uspořádání jsou protokoly SSL/TLS ([4] a [5]), kde se používají tato základní autentizační schémata: varianta s asymetrickou šifrou pro autentizaci serveru vůči klientovi a varianta s digitálním podpisem pro autentizaci klienta vůči serveru (volitelně).

Nespoléhejme na systém

Možná si teď kladete otázku, proč se má návrhář zabývat extra autentizačním protokolem, když většina komunikačních platforem už nějakou autentizaci poskytuje. Pokud by se „to“, co tyto systémy často považují za autentizaci, dalo tímto termínem skutečně nazvat, pak by otázka byla jistě na místě. Bohužel tomu tak v řadě případů není a architektům nezbytvá

než tento problém vyřešit na vyšší úrovni a po svém. Pro přiblížení si připomeňme slabinu, která číhá v prostředí, kde si ji řada uživatelů dodnes téměř neuvědomuje. Tím prostředím je služba krátkých textových zpráv (SMS) v sítích GSM, kde drtivá většina uživatelů stoprocentně věří tomu, že přijatá zpráva byla odeslána ze zařízení, jehož číslo je zobrazeno na displeji jejich telefonu. Dokonce se v praxi setkáme i s různými dálkově ovládanými moduly, které ochotně vyplní příkazy došlé v otevřené textové zprávě. Stačí jen, když tato zpráva pochází ze „správného“ čísla. To vše se přitom děje na pozadí toho, že už od roku 2001 existuje snadno dostupná utilita [3], která útočníkovi umožňuje odesílat krátké textové zprávy s prakticky libovolnou identitou odesílatele. Program je sice napsán pro nepříliš rozšířenou platformu Palm PC, avšak přepsání volně dostupného zdrojového kódu pro libovolné prostředí (včetně javových aplikací pro mobilní telefony) je jistě jen otázkou času a nálady. Tento stav je primárně důsledkem kostrbaté koncepce autentizace v sítích GSM, která již nedokáže pokrýt určité specifické protokoly. Toho pak dovedně využívají „žertovné“ aplikace typu [3]. Architektům a uživatelům zase nezbytvá než o zranitelnosti vědět a řešit vznikající rizika po svém. Například pomocí metod přiblížených v tomto článku.

Vlastimil Klíma, Tomáš Rosa,
klíma@lec.cz, trosa@ebanka.cz

LITERATURA

- [1] Hwang, Y.-H., Yum, D.-H., and Lee, P.-J.: EPA: An Efficient Password-Based Protocol for Authenticated Key Exchange, in Proc. of ACISP 2003, pp. 452-463, 2003
- [2] Menezes, A.-J., van Oorschot, P.-C., and Vanstone S.-A.: Handbook of Applied Cryptography, CRC Press, 1997
- [3] Projekt SMS spoof, <http://www.waste.org/~terje/palm/SMSspoof/>, viz také <http://www.securityteam.com/tools/50P001P5GK.html>
- [4] Rescorla, E.: SSL and TLS: Designing and Building Secure Systems, Addison-Wesley, New York, 2000
- [5] RFC 2246: Allen, C. and Dierks, T.: The TLS Protocol, v. 1.0, January 1999
- [6] E-archivy <http://cryptography.hyperlink.cz> a <http://crypto.hyperlink.cz>