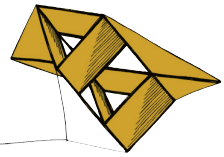


Bezpečné předávání zpráv s použitím blockchainu

Martin Procházka[1]

1 Motivace

Na zabezpečení komunikace v prostředí internetu i mimo něj je v dnešní době kladen obrovský důraz. Následky prozrazení obsahu komunikace mohou být v mnohých případech těžko napravitelné až fatální. Tato problematika byla historicky více či méně úspěšně řešena, cílem této práce bylo se s těmi úspěšnějšími řešeními seznámit a navrhnout řešení, které bude v některých aspektech lepší, než ta stávající. Jako prostředek pro překonání nedostatků v současných systémech bude sloužit decentralizovaný blockchain s podporou chytrých kontraktů, tedy programů spouštěných decentralizovaně bez nutnosti vkládat důvěru v konkrétní uzel sítě. Mezi zkoumané protokoly patří PGP, Signal a v rámci přenosu důvěry také TLS.

2 Komunikační protokol

Způsob jakým komunikace probíhá je definován komunikačním protokolem. Aby bylo možné protokol považovat za bezpečný, musí mít následující vlastnosti: důvěrnost – odeslanou zprávů dokáže číst výhradně adresát; integrita – jakékoliv modifikace odeslané zprávy lze opravit nebo detekovat; autentikace – identita odesilatele je ověřena a odeslaná zpráva je s identitou odesilatele přímo, či nepřímo, svázána; odmítnutelnost – třetí strana nedokáže prokázat, kdo je ve odesilatelem zprávy. Tyto vlastnosti garantují bezpečný přenos do chvíle, než jednomu z účastníků komunikace uniknou stavové proměnné protokolu. Pokud ke kompromitaci dojde, je snaha omezit její dopad, proto jsou navíc žádoucí další dvě vlastnosti: dopředné zabezpečení – uniklé stavové informace nedostačují k prolomení historické komunikace; a zpětné zabezpečení – protokol se z úniku dokáže zotavit a obnovit důvěrnost komunikace. Všechny tyto vlastnosti nabízí protokol Signal, který byl zvolen jako stavební kámen blockchainové implementace.

3 Autentikace

Uživatelé vystupují pod různými identitami (emailové adresy, telefonní čísla, . . .), které je nutné ověřovat. Pro tento účel v systému figurují verfikační autority. Ověření funguje na bázi předání tokenu přes ověřovaný kanál a blockchainovém *commit & reveal* schématu. Navrženy byly dva protokoly, v první variantě chytrý kontrakt přiřazuje verifikační autoritu nedeterministicky, v druhé variantě si uživatel sám vybírá, u které verifikační autority se ověří. Tyto protokoly jsou dále analyzovány z pohledu bezpečnosti a implementovatelnosti na blockchainu. Diskutován je také způsob, jakým jsou autority přidávány do systému.

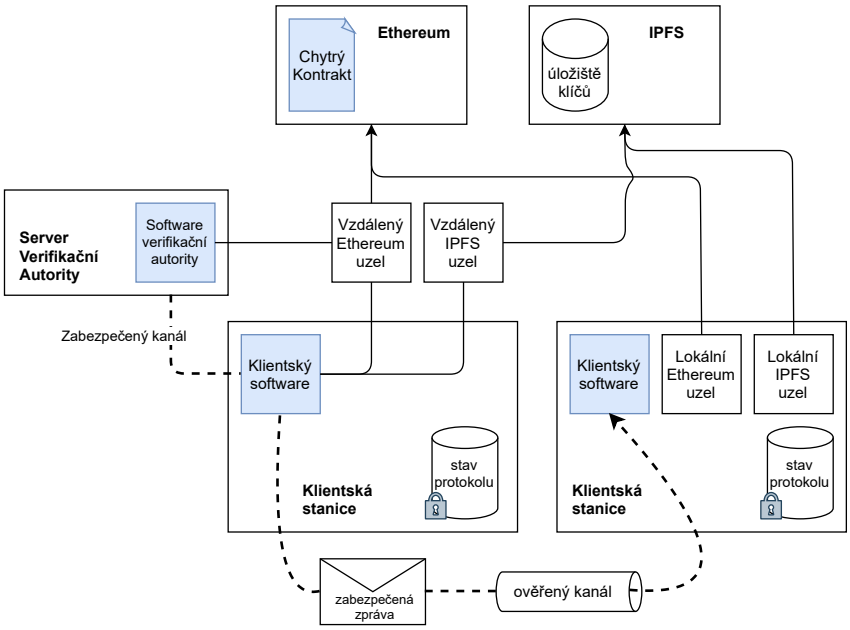
[1] student bakalářského studijního programu Inženýrská informatika, obor Informatika, e-mail: prochazm@students.zcu.cz

4 Distribuce klíčů

Konverzace v Signalu začíná výměnou klíčů, která je realizována protokolem X3DH (Marlinspike (2017)). Ten vyžaduje, aby měl každý uživatel jeden klíč identitní, střednědobý a jednorázový – ten je zahájením relace spotřebován. Jednorázových klíčů je pochopitelně nutné uchovávat větší množství, jinak by mohlo dojít k jejich vyčerpání. Ukládání klíčů přímo na blockchain je kvůli cenovým implikacím prakticky nerealizovatelné, řešením je ukládat tyto klíče na IPFS (Benet (2014)). Na blockchainu je pak uložena pouze adresa souboru s klíči uloženého na IPFS. Tím je dosaženo konstantní ceny za uložení sady veřejných klíčů nezávisle na velikosti n-tice jednorázových klíčů.

5 Realizace

Výsledný produkt se skládá ze tří programových celků – sada chytrých kontraktů, klientská aplikace a server verifikační autority. Chytré kontrakty jsou psány v jazyce Solidity pro blockchain sítě Ethereum (Wood (2020)), ostatní programové vybavení je řešeno na platformě Node.js. Pohled na architekturu navrženého systému je vidět na obrázku 1.



Obrázek 1: Architektura systému, realizovaný software je zvýrazněn modře.

Literatura

Marlinspike, M. The X3DH Key Agreement Protocol. Technical report, November 2016. Dostupné z: `https://www.signal.org/docs/specifications/x3dh/x3dh.pdf`.

Wood, G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper. 2020, 3e2c089. Dostupné z: `https://github.com/ethereum/yellowpaper`.

Benet, J. IPFS - Content Addressed, Versioned, P2P File System. Dostupné z: `https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf`.