VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

ŠKÁLOVATELNÉ HLASOVÁNÍ S OCHRANOU SOUKROMÍ HLASŮ ZALOŽENÉ NA BLOCKCHAINU

SCALABLE 1-OUT-OF-K BLOCKCHAIN-BASED VOTING WITH PRIVACY OF VOTES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. IVANA STANČÍKOVÁ**
AUTHOR

VEDOUCÍ PRÁCE **Ing. IVAN HOMOLIAK, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2020/2021

Zadání diplomové práce

Studentka: **Stančíková Ivana, Bc.**
Program: Informační technologie
Obor: Kybernetická bezpečnost
Název: **Škálovatelné hlasování s ochranou soukromí hlasů založené na blockchainu**
Scalable 1-out-of-k Blockchain-Based Voting with Privacy of Votes
Kategorie: Bezpečnost
Zadání:

1. Nastudujte a srovnejte existující (distribuované) protokoly pro elektronické hlasování s ohledem na ochranu soukromí, utajení hlasovaní, možnost ověření výsledků nezávislou stranou, odolnost proti selhání, verifikovatelnost a škálovatelnost.
2. Nastudujte principy fungování blockchainu a platforem pro smart kontrakty.
3. Navrhněte škálovatelné řešení elektronického hlasování založené na smart kontraktech, zajišťující maximální možnou ochranu soukromí hlasujících a zároveň podporující zotavení po chybě.
4. Vytvořte implementaci demonstrující funkčnost navrženého řešení na alternativní platformě pro smart kontrakty (jiné než Ethereum).
5. Zhodnoťte navrženou implementaci s ohledem na výkonnost a náklady. Diskutujte dosažené výsledky v kontextu celostátních voleb.
6. Porovnejte své řešení s existujícími přístupy a diskutujte možná rozšíření.

Literatura:

- Hao, Feng, Peter YA Ryan, and Piotr Zieliński. "Anonymous voting by two-round public discussion." *IET Information Security* 4.2 (2010): 62-67.
- A. Kiayias and M. Yung. Self-tallying elections and perfect ballot secrecy. In International Workshop on Public Key Cryptography, pages 141-158. Springer, 2002.
- Baudron, Olivier, et al. "Practical multi-candidate election system." *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*. 2001.
- McCorry, Patrick, Siamak F. Shahandashti, and Feng Hao. "A smart contract for boardroom voting with maximum voter privacy." *International Conference on Financial Cryptography and Data Security*. Springer, Cham, 2017.
- Groth, Jens. "Efficient maximal privacy in boardroom voting and anonymous broadcast." *International Conference on Financial Cryptography*. Springer, Berlin, Heidelberg, 2004.
- Lueks, Wouter, Iigo Querejeta-Azurmendi, and Carmela Troncoso. "VoteAgain: A scalable coercion-resistant voting system." *arXiv preprint arXiv:2005.11189* (2020).

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Homoliak Ivan, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 11. listopadu 2020

Abstrakt

Práce se zabývá elektronickými systémy pro hlasování z pohledu jejich vlastností a požadavků, které jsou na ně kladeny. Cílem práce je vytvoření protokolu pro elektronické hlasování, který splňuje požadavky na ochranu soukromí hlasujících, je škálovatelný a odolný proti selhání. Práce zkoumá již existující protokoly a srovnává je dle dosažených vlastností. Vytvořené řešení, využívající smart kontraktů na blockchainu, kombinuje přístupy ze zkoumaných protokolů pro dosažení požadovaných vlastností. Škálovatelnosti dosahuje díky rozdělení hlasování do několika smart kontraktů, z nichž každý realizuje hlasování pouze pro určitou část hlasujících, a následné agregaci dílčích výsledků. Práce se také zabývá problémem nalezení vhodné platformy pro implementaci navrženého systému.

Abstract

The main subject of this work is the assessment of electronic voting systems with regard to their required and achieved properties. The goal of this project is designing an electronic voting protocol that satisfies the requirements for privacy protection while also being scalable and fault-tolerant. Existing protocols are examined and compared according to their properties. The design proposed in this work uses smart contracts on blockchain and combines the approaches from the examined solutions. Scalability is achieved by dividing the process of voting between several smart contracts. Each of these contracts carries out the voting in small scale with only a subset of voters and the partial results are then aggregated. The problem of finding a suitable platform for implementation of the proposed protocol is also addressed in this work.

Klíčová slova

elektronické hlasování, internetové hlasování, e-voting, blockchain, smart kontrakt

Keywords

electronic voting, internet voting, e-voting, blockchain, smart contract

Citace

STANČÍKOVÁ, Ivana. *Škálovatelné hlasování s ochranou soukromí hlasů založené na blockchainu.* Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ivan Homoliak, Ph.D.

Škálovatelné hlasování s ochranou soukromí hlasů založené na blockchainu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Ivana Homoliaka, Ph.D. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Ivana Stančíková
15. května 2021

Poděkování

Ráda bych poděkovala vedoucímu mé práce Ing. Ivanu Homoliakovi, Ph.D. za odborné vedení, konzultace a cenné rady.

Obsah

Kapitola 1

Úvod

Tradiční způsob organizace voleb nebo referend spoléhá na papírové hlasovací lístky a nemalé lidské zdroje věnované zajištění průběhu hlasování i následného sčítání hlasů. Pro hlasující je až na výjimečné případy jedinou možností účasti na hlasování osobní přítomnost na konkrétním místě, kde je nutné zajistit kontrolu, zdali již daný volič lístek neodevzdal, či je-li vůbec oprávněn se účastnit. Tato forma hlasování, ač stále široce využívána, se v době digitálních technologií zdá být nepraktická. Manuální zpracování hlasovacích lístků nechává značný prostor pro pochybení (ať už neúmyslné, či způsobené snahou ovlivnit průběh nebo výsledek hlasování), je nákladné a pomalé.

Stejně jako v mnoha jiných odvětvích lidské činnosti se proto i zde nabízí možnost využití informačních technologií pro vylepšení tohoto systému a přizpůsobení jeho vlastností moderní době. Ačkoli převedení tradičního volebního systému do elektronické podoby skutečně může mít pozitivní přínos, zároveň přináší řadu nových problémů, které jsou velmi specifické pro tuto aplikaci.

Elektronické hlasování je v současnosti velmi aktuálním tématem a proto také velmi aktivním odvětvím výzkumu. Cílem je vytvoření systému, který by nejen přinesl výhody oproti tradičnímu přístupu, jak pro autoritu organizující hlasování, tak pro samotné účastníky, ale aby zároveň splňoval všechny požadavky kladené na hlasování obecně. Musí především chránit soukromí hlasujících, zajistit korektní spočtení všech hlasů a umožnit ověření správnosti průběhu i výsledku hlasování. Současné přístupy často nesplňují všechny požadavky, které jsou na ně kladeny, z čehož při nasazení v praxi plyne i nedůvěra hlasujících.

Jednou z aktuálně velmi skloňovaných a rychle se rozvíjejících technologií je blockchain. Ten již neslouží pouze jako databáze transakcí kryptoměn. Jednou z možností využití blockchainu jsou smart kontrakty, pro které jsou uzly sítě blockchainu distribuovanou výpočetní platformou. Vlastnosti blockchainu a smart kontraktů ve spojení s homomorfní kryptografií mohou posloužit jako platforma pro elektronický hlasovací systém.

V kapitole 2 této práce je podrobněji vysvětlen koncept elektronického hlasování a základní přístupy k jeho realizaci. Zároveň jsou zde vymezeny konkrétní požadavky kladené na elektronické hlasovací systémy. Již existujícími přístupy a protokoly pro elektronické hlasování se zabývá kapitola 3. Tyto přístupy jsou srovnány dle jejich schopností vyhovět požadavkům definovaným v kapitole 2. Kapitola 4 se soustředí na existující blockchainové platformy, které umožňují vytváření smart kontraktů. V kapitole 5 je představeno možné řešení elektronického hlasování založené právě na smart kontraktech, a dále je v kapitole 6 popsána vytvořená implementace tohoto návrhu. Výsledný systém je zhodnocen z hlediska výkonnosti i nákladů v kapitole 7.

Kapitola 2

Elektronické hlasování a jeho vlastnosti

Pojem elektronické hlasování [13], anglicky e-voting, zastřešuje množství různých způsobů realizace voleb využívajících informačních technologií. Samotná definice toho, co vlastně je považováno za elektronické hlasování není jednotná a postupem času se měnila. Jako elektronické tak bývá označováno hlasování, kde je pouze odevzdání hlasovacího lístku nebo pouze sečtení hlasů prováděno s využitím elektronických prostředků, ale i hlasování využívající informačních technologií pro celý proces (tedy odevzdání i sčítání hlasů).

2.1 Rozdílná pojetí elektronického hlasování

Budeme-li uvažovat takto široké vymezení, můžeme tyto elektronické hlasovací systémy rozdělit do dvou kategorií podle toho, zda spoléhají na specializovaná nebo víceúčelová zařízení.

Jedním ze zařízení určených přímo pro využití na hlasovacích stanovištích je tzv. DRE (Direct Recording Electronic) [15], terminál realizující interakci s hlasujícím i správu lístků. Hlasovací lístek je v tomto případě elektronický záznam uložený na paměťovém médiu zařízení, který byl hlasujícím zadán pomocí např. dotykového displeje. DRE tedy vůbec nepracuje s papírovými volebními lístky, může ale nabízet možnost vytištění dokladu pro umožnění kontroly správnosti uložených údajů nebo ruční přepočítání hlasů.

Další možností jsou skenovací zařízení [15]. Pro jejich použití je nezbytné zachování papírových hlasovacích lístků, které jsou předtištěny a následně hlasujícím označeny tak, aby bylo možné jejich strojové zpracování. Samotné naskenování lístku může být prováděno přímo hlasujícím, který tak má možnost zkontrolovat, že byl jeho hlas korektně přijat, nebo až pracovníky, kteří lístky zpracovávají po jejich odevzdání.

Z pohledu této práce jsou však relevantnější ty systémy, které nevyžadují specializované zařízení. Tento přístup předpokládá, že hlasující může svůj lístek odevzdat s využitím libovolného zařízení - například osobního počítače nebo i mobilního telefonu. Toto zařízení může být umístěno na hlasovacím stanovišti, pod kontrolou autority organizující hlasování, daleko zajímavější je však pro účastníky hlasování možnost odevzdat svůj hlas ze svého vlastního zařízení a odkudkoli. Tento přístup bývá označován jako internet voting [46]. Dále v této práci však bude právě tento přístup označován termíny elektronické hlasování nebo e-voting.

Prozatím nebyl ani u jedné z možností zmíněn způsob ověření identity účastníků a jejich oprávnění se účastnit hlasování. Tato činnost může být stále prováděna manuálně, nebo být součástí elektronického systému. V každém případě však musí být zajištěno, aby nebylo možné získat informaci o volbě konkrétního hlasujícího, což je, jak bude popsáno dále, jedna z hlavních požadovaných vlastností hlasovacího systému.

2.2 Požadavky na elektronický hlasovací systém

Stejně jako u tradičního způsobu realizace hlasování, i u elektronického hlasování je nutné definovat množinu požadavků, které jsou na systém kladeny. Vymezení toho, které požadované vlastnosti je systém schopen zajistit, pak umožňuje srovnání různých přístupů a určení možností jejich praktického využití.

2.2.1 Ochrana soukromí (angl. privacy)

Ochranou soukromí hlasujícího rozumíme utajení jeho hlasu. Spojit konkrétního hlasujícího s lístkem, který odevzdal tedy nesmí být možné. V souvislosti s ochranou soukromí je nutné také uvažovat, zdali je hlasujícímu poskytnut nějaký doklad o tom, jak hlasoval. Pokud takovýto doklad existuje, je možné, že tato skutečnost bude zneužita k ovlivnění výsledku hlasování například nátlakem na hlasující či kupováním hlasů.

2.2.2 Dokonalé utajení hlasování (angl. perfect ballot secrecy)

Cílem zavedení tohoto požadavku, který definují Kiayias a Yung [26], je posílení ochrany soukromí. V hlasovacím systému dosahujícím dokonalého utajení hlasování není možné odhalit, jak hlasoval jeden konkrétní hlasující, ale ani výsledný součet hlasů určité podmnožiny hlasujících, aniž by se na odhalení hlasů podíleli všichni zbývající hlasující.

2.2.3 Self-tallying

Tato vlastnost [26] hlasovacího systému vymezuje podmínky, za nichž dochází ke sčítání hlasů. V tomto případě mohou sčítání provést nezávisle na sobě jednotliví účastníci hlasování, případně i nezávislá strana dozorující průběh hlasování. Tento přístup tedy nevyžaduje přítomnost důvěryhodné autority provádějící sčítání hlasů, případně je-li tato autorita zavedena, jí zveřejněný výsledek je ověřitelný.

2.2.4 Odolnost proti selhání (angl. fault tolerance)

Spíše než vlastností systému je odolnost proti selhání mírou schopnosti systému vyrovnat se s nekorektním chováním jednoho nebo více účastníků v průběhu hlasování. Nejčastěji předpokládáme nekorektní chování ze strany hlasujícího, jako je odevzdání neplatného lístku nebo neodevzdání lístku žádného, nicméně i autorita, je-li v daném systému uvažována, může průběh hlasování narušit. Při hlasování ve velkém rozsahu (s velkým počtem hlasujících) je nežádoucí opakovat celý proces od začátku v případě, že k nekorektní situaci dojde, ačkoli v menším měřítku toto akceptovatelné být může.

2.2.5 Verifikovatelnost

Pro hlasujícího je splněním tohoto požadavku zajištěna možnost ověřit si, že jím odevzdaný hlas byl započítán do celkového výsledku. Z pohledu korektnosti celého průběhu hlasování

má každý účastník hlasování možnost ověřit, že byly správně sečteny všechny platné hlasy oprávněných voličů.

2.2.6 Škálovatelnost

Řada doposud navržených protokolů (jako např. [24, 26, 28]) uvažuje pouze malý počet hlasujících, ve větším rozsahu by tyto protokoly nemohly být využity pro jejich výpočetní nebo cenovou náročnost. Po škálovatelném hlasovacím systému tak požadujeme, aby byl schopen rozšíření pro potřebný počet hlasujících (i kandidátů) dle nároků praktického využití (např. správní rada společnosti s nejvýše desítkami hlasujících, ale také celostátní volby s miliony voličů).

2.2.7 Bezespornost

Zatímco verifikovatelnost umožňuje ověřit, že účastníci hlasování dodrželi stanovený protokol hlasování, pokud jde o obsah hlasovacích lístků a výpočet výsledků, můžeme požadovat také všeobecnější kontrolu nad chováním účastníků v souladu s protokolem. V tomto případě již neuvažujeme možnost zpětně ověřit, že k nekorektnímu postupu nedošlo, ale přímo požadujeme, aby systém takovou situaci vůbec neumožnil, nebo aby byla systémem detekována v průběhu hlasování. S touto vlastností hlasovacího systému se setkáváme pod názvem bezespornost (angl. dispute-freeness) [26, 22].

2.2.8 Problém ovlivňování hlasujících

Může nastat situace, kdy je hlasující nucen jinou osobou hlasovat pro určitého kandidáta, především probíhá-li hlasování v nekontrolovaném prostředí (z vlastního zařízení) nebo existuje-li doklad o tom, jaký hlas odevzdal. Systémy, které řeší tento problém (označované jako coercion-resistant [27]), obvykle umožňují hlasujícímu odevzdat pod nátlakem neplatný hlas, který může později stáhnout nebo nahradit platným hlasem dle vlastního rozhodnutí, přičemž útočník nemá možnost dozvědět se, zdali tak hlasující učinil, či nikoli.

2.2.9 Problém posledního hlasujícího

Především u protokolů umožňujících self-tallying vzniká situace, kdy má hlasující, který odevzdává svůj lístek jako poslední, možnost odhalit výsledek hlasování ještě předtím, než sám lístek odevzdá. To mu může umožnit změnit svou volbu s ohledem na stav hlasování, což představuje nežádoucí situaci. U self-tallying protokolů musí být tedy také zajištěno, aby žádný (částečný) výsledek hlasování nemohl být odhalen dříve, než všichni účastníci odevzdají své hlasy a hlasování je ukončeno.

Kapitola 3

Srovnání existujících řešení

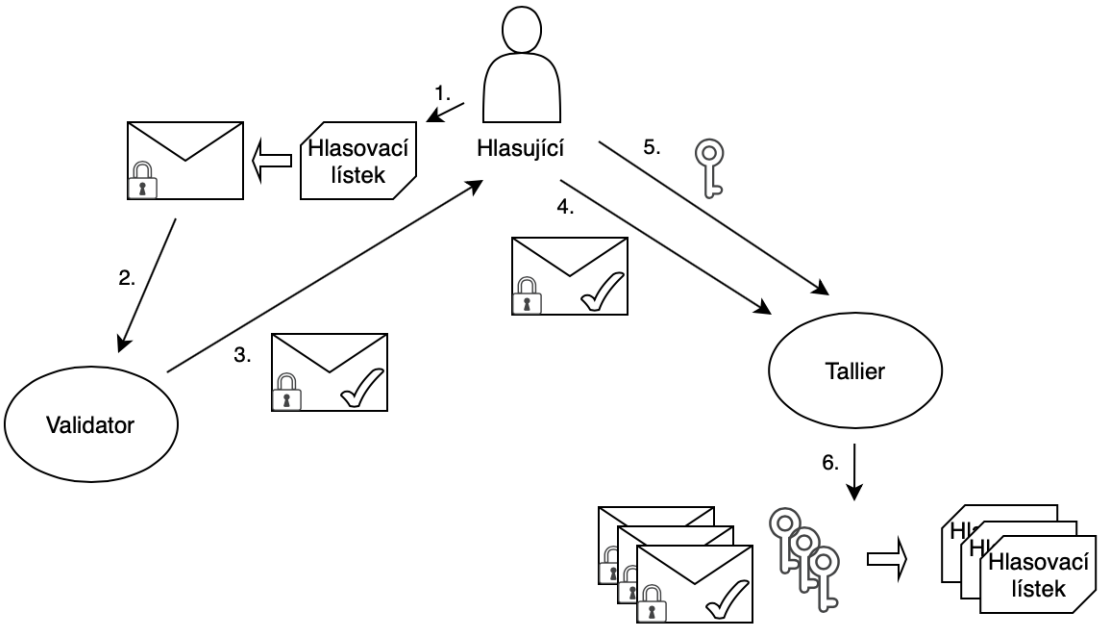
V následujícím přehledu budou popsány doposud navržené přístupy k elektronickému hlasování. Tyto přístupy budou srovnány s ohledem na to, které požadované vlastnosti hlasovacího systému splňují. Můžeme zde pozorovat, že hlavním problémem, kterému čelí autoři těchto protokolů, je především dosažení určitých požadovaných vlastností zároveň. Častými příklady jsou zajištění ochrany soukromí hlasujících významně komplikující verifikovatelnost protokolu a naopak, nebo zajištění dokonalého utajení hlasování, kdy hlasování selže bez účasti všech hlasujících.

3.1 Anonymní hlasující

V prvním z možných přístupů je obsah jednotlivých hlasovacích lístků odtajněn a ochrana soukromí hlasujících je zajištěna tím, že tento hlas v otevřené formě již není možné spojit s hlasujícím, který ho odevzdal. Takovéto protokoly vyžadují, aby komunikace probíhala anonymním komunikačním kanálem. Zástupcem tohoto přístupu je protokol Sensus [11].

3.1.1 Protokol Sensus

Hlasovací systém Sensus [11] využívá principu slepých podpisů [8], díky kterému lze získat podpis zprávy, aniž by podepisující znal její obsah. Systém je tvořen několika moduly s oddělenými funkcemi, z nichž podstatné jsou především dva. Modul *validator* ověřuje podpis hlasujícího a modul *tallier* zodpovídá za sesbírání všech hlasovacích lístků a součet výsledku. Proces hlasování probíhá tak, že hlasující nejprve vytvoří svůj hlasovací lístek, na který následně aplikuje dvě vrstvy šifrování. První z nich zašifruje lístek tajným klíčem hlasujícího, druhá zajistí maskování lístku pro získání slepého podpisu autority. Tento lístek je navíc hlasujícím podepsán a odeslán do modulu *validator*, který ověří, že podpis odpovídá registrovanému hlasujícímu. Pokud je toto ověření úspěšné, *validator* lístek podepíše. Hlasující nyní může odstranit vrstvu šifrování maskující obsah lístku a předat lístek modulu *tallier*. Hlasovací lístek, který obdrží modul *tallier* je tedy šifrovaný klíčem hlasujícího s podpisem ověřujícího modulu. Modul *tallier* v tuto chvíli ví, že obdržel hlasující lístek oprávněného hlasujícího, nezná však jeho obsah potřebný k součtu výsledku. Klíč pro dešifrování lístku poskytne sám hlasující výměnou za doklad, tvořený hlasovacím lístkem, který *tallier* obdržel a sám podepsal. Po ukončení hlasování *tallier* dešifruje všechny hlasovací lístky a kromě výsledku zveřejní také všechny hlasovací lístky v zašifrované i dešifrované podobě spolu s patřičnými klíči.



Obrázek 3.1: Schéma elektronického hlasování s anonymním hlasujícím.

Oddělení modulů *validator* a *tallier* a využití slepých podpisů je zavedeno z důvodu zajištění ochrany soukromí. Systém ale předpokládá, že hlasy nedorazí do obou modulů ve stejném pořadí, což ovšem není schopen zajistit. Pokud by k tomu došlo (např. v důsledku malého počtu hlasujících) mohly by tyto dva moduly společně odhalit identitu hlasujících ve spojení s konkrétními hlasy. Pokud se některý z registrovaných hlasujících nezúčastní hlasování nebo odevzdá neplatný lístek, průběh procesu hlasování to neovlivní, nicméně zde chybí možnost ověření, zda byly skutečně zveřejněny a započítány všechny platné odevzdané hlasy a zda modul *validator* nevložil hlasy za nezúčastněné registrované hlasující. Ověřit přítomnost svého vlastního hlasu mezi zveřejněnými může pouze sám hlasující.

Systém Sensus vychází ze staršího návrhu [20], ten se liší tím, že autorita sčítající hlasy zveřejní pouze zašifrované hlasovací lístky a hlasující poskytnou klíče k dešifrování až poté, co zkontrolují, že jejich hlas je mezi zveřejněnými. Vlastnosti popsané u protokolu Sensus nicméně platí i zde.

Sensus ilustruje postup (Obrázek 3.1), který můžeme s obměnami nalézt v dalších protokolech, jako např. [37, 33]. Společné mají tyto protokoly to, že se primárně soustředí na zajištění ochrany soukromí a verifikovatelnost, nikoli na dokonalé utajení hlasování nebo self-tallying. Nejsou však obvykle narušitelné neúčastí registrovaných hlasujících nebo nevalidními lístky a mají také potenciál dosáhnout dobré škálovatelnosti.

3.2 Systémy založené na homomorfním šifrování

Další přístup k realizaci e-votingu je založen na homomorfním šifrování. Homomorfní šifrování [10] umožňuje provádět početní operace nad zašifrovanými daty bez nutnosti dešifrování jednotlivých položek. Pro zprávy m_1 a m_2 a jim odpovídající šifrované texty c_1 a c_2 takové, že $c_1 = E(m_1)$ a $c_2 = E(m_2)$ můžeme homomorfní vlastnost šifrování E vyjádřit jako

$$c_1 \otimes c_2 = E(m_1 \oplus m_2).$$

Díky tomu nemusí být pro získání výsledku hlasování dešifrovány jednotlivé hlasy a tedy i v případě, kdy je možné konkrétní hlas spojit s hlasujícím, není možné odhalit obsah lístku. Jelikož v tomto případě není odhalen obsah jednotlivých hlasovacích lístků a není tedy nutné oddělit hlasujícího od lístku, který odevzdal, předpokládají takovéto protokoly veškerou komunikaci všech účastníků (hlasujících i autorit) formou broadcastu nebo veřejné vývěsky, tedy tak, aby měl každý účastník přístup k veškeré komunikaci, která během procesu hlasování proběhla.

3.2.1 Protokol dle Benaloh, Yung a Muti

Jedním z prvních zástupců toho přístupu je protokol [5], který zodpovědnost za sčítání hlasů distribuuje mezi několik čítačů a uvažuje výběr pouze ze dvou možností (hlas ano/ne). Před začátkem samotného hlasování probíhá přípravná fáze. Každý z čítačů nejprve vygeneruje a zveřejní dvojici hodnot, dle specifikovaných kritérií, který nazveme parametry hlasování. Každý hlasující pak použije tyto parametry k vytvoření množiny zkušebních hlasovacích lístků a pro každý z nich dokáže jejich validitu pomocí interaktivního důkazu. V dalším kroku je dokázáno, že čítač je schopen rozpoznat volbu na hlasovacích lístcích. Každý z hlasujících na každém zkušebním lístku náhodně označí jednu z možných voleb (ano/ne) a čítač dešifruje výsledek z každé sady lístků. Výsledky dešifrované čítači a hlasujícími jsou porovnány a v případě, že se kterýkoli z nich neshoduje, jsou parametry čítače označeny za nevalidní a hlasování nemůže dále pokračovat. Je-li vše v pořádku, následuje samotné hlasování. Jeden z množiny zkušebních lístků je vybrán jako skutečný lístek pro hlasování, znovu je prokázán jako validní a poté označen skutečnou volbou hlasujícího. Čítač provede částečné sečtení jeho parametrům odpovídající části hlasů a provede důkaz, že jeho výsledek je validní. Součet výsledků jednotlivých čítačů je celkovým výsledkem hlasování.

Tento protokol zajišťuje ochranu soukromí hlasujících v případě, že alespoň jeden z čítačů je poctivý. Ukáže-li se během přípravné fáze, že některý ze čítačů se nechová korektně, je nutné začít znovu od začátku se zbývajícími čítači. Díky veřejnému komunikačnímu kanálu a vyžadovaným důkazům je protokol verifikovatelný. Není však kvůli výpočetní náročnosti vhodný pro velké počty hlasujících, lze jej však rozšířit, tak aby umožňoval volbu mezi více kandidáty.

3.2.2 Protokoly dle Cramer et al.

Snížení výpočetní náročnosti a tedy lepší škálovatelnosti tohoto přístupu dosáhly systémy od Cramer et al. [9, 10]. V hlasovacím protokolu [10] je opět množina autorit, mezi které je rozdělen tajný klíč pro dešifrování výsledku. Hlasující odevzdávají své lísky zašifrované veřejným klíčem autorit na veřejnou nástěnku. Pro získání výsledku musí korektně spolupracovat určitý počet autorit (tedy ne nutně všechny, dle zvolených parametrů systému). Stejný počet nepoctivých autorit by pak byl nutný k odhalení jednotlivého hlasu a tedy narušení soukromí hlasujících. Pro zajištění verifikovatelnosti systému jsou vyžadovány průběžné důkazy validity zveřejňovaných údajů od účastníků (hlasujících i autorit). Nevýhodou z hlediska škálování je nutnost výpočtu výsledku jako prohledávání prostoru možných řešení (t.j. hrubou silou).

3.2.3 Protokol dle Baudron et al.

Rozšířením tohoto přístupu pro více kandidátů a praktické využití v celostátních volbách se zabývali Baudron et al. [4] Systém je kromě hlasujících tvořen také hierarchicky uspořá-

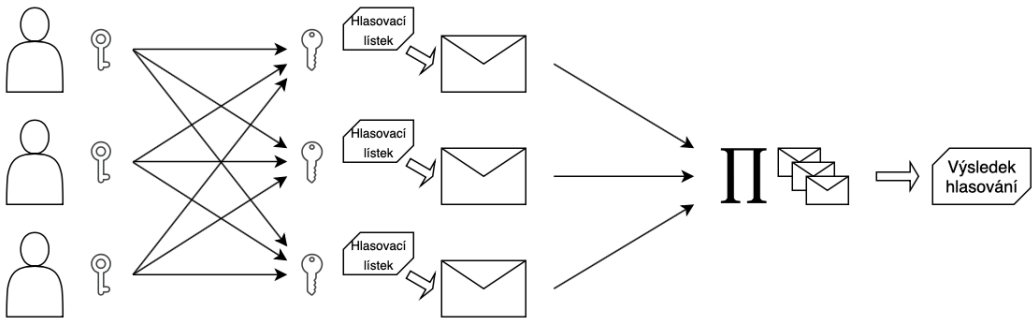
danými autoritami (např. lokální, regionální, celostátní). Každá z nich má svou nástěnku, na kterou do vyhrazeného prostoru zapisují autority nižší nebo přímo voliči v případě nejnižší úrovně. Každá autorita zveřejní svůj veřejný klíč. Hlasující svůj hlas zašifruje klíčem autority z každé úrovně zvlášť a zapíše tyto šifrované hlasy na nástěnku lokální autority spolu s důkazy, že jsou lístky validní a že ve všech je šifrován stejný hlas. Sčítání pak probíhá postupně po jednotlivých úrovních. Nejdříve všechny lokální autority společně dešifrují výsledek hlasů na lokální úrovni. Tyto mezivýsledky zapisují na nástěnku regionální autority. Regionální autority pak provedou stejný proces a navíc zkontrolují zda souhrnný výsledek na jejich úrovni odpovídá výsledku předchozí úrovně. Takto je postupováno až po nejvyšší úroveň. Systém dosahuje menší výpočetní náročnosti oproti předchozím, což umožňuje volby mezi více kandidáty i účast většího počtu hlasujících. Ochranu soukromí zajišťuje, podobně jako v protokolu dle Cramer at al. [10], rozdělení tajného klíče mezi autority na dané úrovni. Pokud je určitý počet autorit poctivý, zbylé nemohou jednotlivý hlas dešifrovat. Výhodou tohoto přístupu je, že umožňuje hlasujícímu registrovat se k hlasování a odevzdat svůj hlas zároveň (během jednoho sezení). V protokolech, které budou popsány dále lze odevzdávání hlasů zahájit až poté, co se zaregistrovali všichni hlasující, což od hlasujících vyžaduje dvě oddělené interakce s hlasovacím systémem.

Všechny doposud zmíněné přístupy vyžadují přítomnost spolehlivých autorit. Pozitivem je, že autorita není pouze jediná a pro znemožnění úspěšného průběhu hlasování nebo narušení soukromí hlasujících je nutné splnění určitých předpokladů (určitý počet neaktivních autorit, nutnost jejich spolupráce). Především z hlediska ochrany soukromí však tyto předpoklady nejsou dostatečně silné. Další přístupy posouvají řešení tohoto problému dál tím, že již přítomnost žádné autoritu nevyžadují.

3.2.4 Protokol dle Kiayias a Yung

Kiayias a Yung [26] zvyšují nároky na ochrany soukromí zavedením pojmu perfect ballot secrecy. Díky tomuto, v kombinaci s principem self-tallying, přenášejí proces sčítání hlasu přímo na hlasující tak, že tento proces může být distribuován mezi účastníky, aniž by mohl kterýkoli z nich odhalit částečný výsledek na podmnožině hlasů. Účast všech hlasujících na sčítání hlasů však není podmínkou, na rozdíl od [38].

Základní schéma tohoto přístupu je ilustrováno na obrázku 3.2. Tento protokol uvažuje grupu G a její generátory g, f, h. Každý z N hlasujících vygeneruje několik dvojic klíčů. Nejprve vytvoří svůj vlastní tajný klíč a_j a veřejný klíč h^{a_j}. Poté vytvoří zbývající dvojice klíčů $s_{i,j}$ a $g^{s_{i,j}}$, $i, j \in 1...N$, z nichž každá odpovídá vždy jednomu ze všech hlasujících. Všechny veřejné klíče jsou zapsány na veřejnou vývěsku. Další postup vede k tomu, že šifrování všech později vložených hlasů bude provázáno tak, že celkový výsledek nebude možné získat, pokud by některý z hlasů chyběl. Veřejný klíč každého hlasujícího h^{a_j} zašifrují všichni hlasující, každý odpovídajícím klíčem $s_{i,j}$. Dešifrováním součinu všech hodnot $(h^{a_j})^{s_{i,j}}$, $i \in 1...N$ svým tajným klíčem a_j získá každý hlasující hodnotu, kterou použije k zašifrování svého hlasu. Hlasovací lístek B_j má pak tvar $h^{s_{1,j}+...+s_{N,j}} f^{v_j}$. Možnosti volby v_j jsou opět pouze dvě, nicméně autoři popisují i možnost rozšíření pro více kandidátů. Výsledek hlasování $\sum_{j=1}^{N} v_j$ pak může vypočítat kterýkoli z účastníků nalezením řešení rovnice $\prod_{j=1}^{N} B_j = f^{\sum_{j=1}^{N} v_j}$. Součet hlasů nemůže být větší než počet hlasujících, tedy dostatečně malý na to, aby bylo možné vypočítat diskrétní logaritmus. Toto je však možné pouze v případě, že své hlasy odevzdali všichni hlasující, kteří se účastnili přípravné fáze protokolu. Autoři protokolu uvažují možnost opravy v případě neúčasti některých hlasu-



Obrázek 3.2: Schéma elektronického hlasování využívajícího homomorfní šifrování s vlastností self-tallying.

jících. Jelikož je ale systém vhodný především pro malý počet hlasujících, předpokládá, že případné opakování protokolu je přijatelné. Protokol vyžaduje důkazy znalosti tajných klíčů. Podobně jako v předchozích protokolech, i zde je verifikovatelnost založena právě na důkazech znalosti tajných klíčů a správnosti zašifrovaných hodnot v jednotlivých krocích.

3.2.5 Protokoly zefektivňující přístup dle Kiayias a Yung

Protokoly dle Hao et al. [24] a Groth [22] se snaží zefektivnit tento princip, při zachování dokonalého utajení hlasování, vlastnosti self-tallying a verifikovatelnosti. Protokol [22] snižuje počet operací umocnění, které musí vypočítat každý hlasující za cenu toho, že probíhá ve více krocích. Pro zahájení hlasování zde každý hlasující j umístí na vývěsku svůj veřejný klíč. Dále nejsou, na rozdíl od [26], na vývěsku umisťovány jednotlivé hlasovací lístky, ale vždy průběžný výsledek, resp. součin doposud odevzdaných lístků. V každém dalším kroku tak vždy hlasuje pouze jeden hlasující. Na pořadí hlasujících zde nezáleží, nicméně každý hlasující může hlasovat až po tom, co předchozí svůj krok dokončí. Hlasující, na kterého přišla řada postupuje tak, že vytvoří svůj hlasovací lístek, vynásobí ho s průběžným výsledkem a svým tajným klíčem odstraní jednu vrstvu šifrování výsledku. Mezivýsledek po svém kroku umístí opět na vývěsku. Poslední hlasující odstraní poslední vrstvu šifrování a z jeho záznamu na vývěsce je opět výpočtem diskrétního logaritmu zjistitelný konečný součet hlasů.

Protokol dle Hao et al. [24] se naopak snaží snížit počet kroků protokolu, a to na pouhé dva. V prvním z nich jsou zveřejněny veřejné klíče g^{x_i} jednotlivými hlasujícími P_i spolu s důkazy o znalosti odpovídajících tajných klíčů x_i. Následně každý hlasující vypočítá g^{y_i} jako $g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} / \prod_{j=i+1}^{N} g^{x_j}$. V druhém kroku jsou zveřejněny samotné hlasovací lístky ve tvaru $g^{x_i y_i} g^{v_i}$ opět s důkazem, tentokrát o korektnosti v_i (zde opět jedna ze dvou možností ano/ne, resp. 0 nebo 1). Výsledek hlasování $\sum_{i=1}^{N} v_i$ může díky vlastnosti self-tallying spočítat kterýkoli účastník jako $\prod_{i=1}^{N} g^{x_i y_i} g^{v_i} = g^{\sum_{i=1}^{N} v_i}$.

Oba protokoly [24] a [22] jsou určené pro hlasování malého počtu hlasujících s volbou mezi dvěma kandidáty. Podařilo se jim však splnit požadavky ochranu soukromí i dokonalé utajení hlasování, mají vlastnost self-tallying a jsou verifikovatelné. Oba protokoly předpokládají, že při malém počtu hlasujících je akceptovatelné hlasování opakovat v případě selhání.

Tyto přístupy (s vlastností self-tallying) musí řešit problém posledního hlasujícího, tedy situaci, kdy poslední hlasující nejprve zjistí výsledek hlasování, a až poté se rozhodne, jaký hlas sám odevzdá. Řešením tohoto problému je nejčastěji zavedení autority uzavírající proces hlasování. Tato autorita ukončí odevzdávání hlasů vložením svého lístku, který nijak neovlivní výsledek. Posledním hlasujícím je tak autorita a všichni účastníci, jejichž hlasy jsou započítány, hlasují za stejných podmínek. Ačkoli toto řešení opět zavádí roli autority, na rozdíl od dřívějších přístupů, tato autorita nemůže nijak ohrozit ochranu soukromí ani sčítání hlasů, její role je omezena pouze na řešení tohoto problému. Tento přístup využívají protokoly [26] a [22].

	OchS	UtHl	S-T	SlhO	Vrf	Škl
Sensus [11]	✓	×	×	✓	×	✓
Benaloh et al. [5]	✓	×	×	×	✓	×
Cramer et al. [9, 10]	✓	×	×	✓	✓	×
Baudron et al. [4]	✓	×	×	✓	✓	✓
Kiayias a Yung [26]	✓	✓	✓	×	✓	×
Hao et al. [24]	✓	✓	✓	×	✓	×
Groth [22]	✓	✓	✓	×	✓	×

Tabulka 3.1: Srovnání vlastností hlasovacích protokolů. OchS = ochrana soukromí, UtHl = dokonalé utajení hlasování, S-T = self-tallying, SlhO = odolnost proti selhání, Vrf = verifikovatelnost, Škl = škálovatelnost.

3.2.6 Protokoly využívající blockchain

Protokoly splňující vlastnosti dokonalého utajení hlasování a self-tallying vyžadují veřejnou vývěsku, nicméně nedefinují jak by měla být realizována. Groth [22] navrhuje jako možnost centrální server, který by komunikaci zajišťoval. Jediný server však představuje možný bod selhání systému. Pokud by tento server nebyl v provozu, hlasování by vůbec nemohlo proběhnout. Uvažovat bychom také museli nekorektně postupující server snažící se ovlivnit průběh hlasovaní.

Protokol **Open Vote Network** od McCorry et al. [28] následuje řešení dle Hao et al. [24], ale vývěsku umísťuje na blockchain. Má podobu smart kontraktů, kromě vývěsky tak získává i výpočetní platformu.

Před začátkem hlasování je úkolem autority ustanovit seznam oprávněných hlasujících a definovat, do kdy musí proběhnout jednotlivé kroky protokolu. V následujících dvou krocích stejně jako v protokolu [24] probíhá výpočet klíčů pro šifrování hlasů a dále samotné odevzdání hlasovacích lístků. V průběhu jsou od hlasujících vyžadovány důkazy znalosti tajného klíče a korektního obsahu hlasovacího lístku. Po ukončení odevzdávání hlasovacích lístků je na blockchainu provedeno sečtení hlasů.

Systém je tvořen dvěma smart kontrakty, z nichž jeden je vyhrazen pro zajištění průběhu jednotlivých kroků hlasování a druhý implementuje kód pro výpočty důkazů. Ačkoli protokol neuvažuje možnost zotavení po chybě, motivuje hlasující k účasti vložením finančních prostředků. Každý hlasující při registraci odevzdá zálohu, o kterou přijde, pokud se následně odmítne hlasování zúčastnit. Protokol je určen pro Ethereum, kde jeho škálovatelnost limituje maximální velikost bloku.

Seifelnasr et al. [39] navrhli variantu hlasovacího protokolu Open Vote Network [28] s lepší škálovatelností. Té dosahují přesunutím větších objemů dat (seznamu hlasujících) a výpočtu výsledku mimo smart kontrakt. Ten pak pouze ověřuje, že výsledek vypočítaný mimo blockchain je správný.

Další vylepšení elektronického hlasování na blockchainu představuje protokol **BBB-voting** [42]. Jak Open Vote Network [28], tak Seifelnasr et al. [39] uvažují pouze dvě možnosti volby (ano/ne, příp. dva kandidáti). BBB-voting umožňuje výběr z libovolného počtu kandidátů, který je, stejně jako počet hlasujících, limitován pouze výpočetní náročností. Implementace však zahrnuje několik optimalizací, díky kterým umožňuje účast více hlasujících než předchozí řešení. Dále je tento systém schopný vypořádat se s neúčastí několika registrovaných hlasujících bez nutnosti opakování celého hlasování.

3.3 Elektronické hlasování v praxi

Několik elektronických hlasovacích systémů bylo již možné vyzkoušet i v praxi. Prvním takovým systémem zajišťujícím verifikovatelnost byl **Helios** [1]. Zdrojový kód Heliosu je dostupný jako open-source a hlasování je možné spravovat prostřednictvím webových stránek [25]. Je však určen pro hlasování v malém měřítku (např. společenství vlastníků), kde nedochází k ovlivňování voličů. Systém vyžaduje, aby správce hlasování nejprve zaregistroval všechny účastníky na centrálním serveru, který zprostředkovává celý průběh hlasování. Zaregistrovaní účastníci obdrží e-mail se všemi údaji potřebnými pro jejich autentizaci a zašifrování hlasovacího lístku. Jakmile je hlasování zahájeno, server přebírá zašifrované hlasovací lístky od uživatelů a zveřejňuje na vývěsce identifikační údaje hlasujících spolu se zašifrovanou formou jejich hlasovacích lístků, což umožňuje hlasujícím zkontrolovat, že jejich lístek byl správně vložen. Po ukončení odevzdávání hlasů jsou zašifrované lístky promíchány a dešifrovány pro sečtení výsledku. Na podobné využití jako Helios cílí i množství komerčních systémů. Jako příklady lze uvést systémy eBallot [14] nebo VoxVote [44].

Mezi systémy dostupnými pro praktické využití najdeme i zástupce hlasování založeného na blockchainu. Jedním z nich **Follow My Vote** [19], systém, který je nabízen i pro hlasování s většími nároky na zabezpečení systému, jako jsou volby v rámci politických stran nebo do veřejných funkcí. Blockchain slouží jako vývěska pro uchování hlasovacích lístků a on-line identit hlasujících. Identifikační údaje na blockchainu nejsou skutečnými osobními údaji uživatele, ty zná pouze autorita organizující hlasování. Autorita zodpovídá za ověření, že každá on-line identita je unikátní pro uživatele oprávněného hlasovat. Follow My Vote navíc umožňuje hlasujícím svůj odevzdaný lístek změnit, případně jej úplně stáhnout.

Podobným systémem je **Voatz** [43]. Ten umožňuje hlasování pomocí aplikace na mobilním zařízení. Identita hlasujícího je ověřena pomocí biometrického systému, který je na daném zařízení dostupný. Poté, co hlasující odevzdá svůj elektronický hlasovací lístek, obdrží doklad o svém hlasu, který může později porovnat ze záznamem na blockchainu pro ověření, zda byl jeho hlas vložen správně. Voatz byl, jako první systém tohoto typu, využit ve federálních volbách v USA, a to v roce 2018 v Západní Virginii [35]. Používání Voatz však zde bylo ukončeno poté, co byl tento systém předmětem studie provedené na Massachusettském technologickém institutu [40]. Autoři studie analyzovali mobilní aplikaci Voatz pomocí reverzního inženýrství a poukázali na řadu bezpečnostních nedostatků (možné narušení soukromí hlasujících i ovlivnění výsledků hlasování útočníkem).

Kapitola 4

Technologie blockchainu a smart kontrakty

4.1 Blockchain

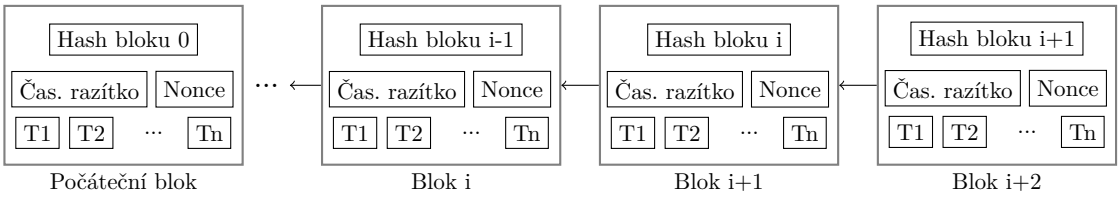
Koncept technologie blockchainu byl poprvé prakticky využit v roce 2008 jako základní mechanismus fungování kryptoměny Bitcoin, jejíž autor (nebo skupina autorů) je znám pod pseudonymem Satoshi Nakamoto [29]. Blockchain můžeme definovat jako databázi všech transakcí, které mezi uživateli systému proběhly. Transakce jsou uloženy v na sebe navazujících blocích. S přibývajícími transakcemi přibývají také nové bloky a cílem je, aby jednou přidaný blok (a v něm obsažené transakce) již nebylo možné změnit ani odebrat. Blockchain je decentralizovaný, kde celá databáze transakcí je distribuována mezi jednotlivými uzly. Každý uzel sítě obsahuje úplnou nebo částečnou kopii řetězce bloků. Kterýkoli držitel kopie řetězce je schopen iterativně dohledat celou historii transakcí.

Blok a transakce

Obrázek 4.1 znázorňuje podobu blockchainu. Každý blok je tvořen [48] hlavičkou a daty transakcí. Důležitou položkou v hlavičce bloku je hodnota hash předchozího bloku v řetězci, která zajišťuje jednoznačné pořadí bloků a také zabraňuje změnám v blocích (v případě jakékoliv změny v bloku by hash hodnota bloku nebyla shodná s hodnotou v bloku následujícím). Dalšími položkami v hlavičce bloku jsou hodnota *nonce* (její účel bude vysvětlen dále), velikost bloku, časové razítko a případně další údaje dle konkrétní implementace. Dále jsou v bloku uloženy záznamy o transakcích. Transakcí bývá nejčastěji převod finančních prostředků mezi uživateli, může se ale jednat o záznam jiné aktivity. Pro ověření transakcí [49] je využívána asymetrická kryptografie. Tajný klíč slouží k podepsání odchozí transakce. Veřejný klíč je znám všem uzlům v síti a ty tak mohou ověřit, že odesílatel transakce skutečně disponuje příslušným tajným klíčem. Transakce v jednom bloku jsou považovány za uskutečněné ve stejný okamžik [12], byť mohly ve skutečnosti dorazit k různým uzlům sítě v různém pořadí.

Vytváření nových bloků

Uzly sítě, které se snaží o vytváření nových bloků se nazývají těžaři (angl. miners). Jejich motivací je finanční odměna, kterou obdrží za každý blok, který do řetězce připojí. Vytvořit nový blok a chtít jej připojit do řetězce může několik těžařů zároveň. O tom, který z těžařů



Obrázek 4.1: Schéma blockchainu. [49]

svůj blok připojí rozhoduje tzv. proof-of-work [32]. Těžaři mají za úkol nalézt takovou hodnotu *nonce*, aby výsledný hash bloku a *nonce* byl menší než určitá hodnota. Tato hodnota může být zvolena podle toho, jaká obtížnost hledání je žádoucí, jelikož právě obtížností je možné ovlivnit, jak často budou do řetězce přibývat nové bloky. Přestože nalezení vhodné hodnoty *nonce* je výpočetně náročné, ověření její správnosti je snadné, vyžaduje pouze výpočet hash funkce s danou *nonce*. Princip takovéto matematické hádanky snižuje pravděpodobnost, že dojde k připojení více bloků ve stejný okamžik. Pokud k tomu přeci jen dojde, řetězec se rozdělí do dvou (případně více) větví [49]. Nejdelší z větví se po určité době stává hlavním pokračováním řetězce, k ostatním pak již další bloky připojovány nejsou.

Alternativním přístupem k rozhodování o připojení konkrétních bloků je tzv. proof-of-stake [48]. Tento princip eliminuje výpočetní náročnost hledání hodnot *nonce*. Kritériem pro výběr uzlu, který připojí další blok, je množství měny, které v systému investoval. Předpokládá se, že uživatel, který vložil dostatečné množství prostředků má zájem na korektním fungování systému. Důsledkem může být, že nové bloky připojuje jen několik nejbohatších uzlů. Často je proto princip proof-of-stake kombinován s dalšími technikami, aby vytváření bloků bylo rovnoměrněji rozloženo mezi uzly sítě. Hodnota *nonce*, pokud se v hlavičce bloku vyskytuje, mívá jiné využití než v případě proof-of-work.

Typy blockchainu

Nejčastěji bývá uvažován veřejný blockchain (označován jako permissionless [48]), k jehož obsahu má přístup kdokoli a kdokoli se také může zapojit do vytváření nových bloků. Není zde přítomna žádná autorita, která by o přístupu k blockchainu rozhodovala. Adresa uživatele v rámci sítě není nutně nijak provázána s jeho skutečnou identitou. Uživatelům tak bývá umožněno zůstat anonymní. Pro dosažení důvěryhodnosti takového blockchainu je nutné využívat techniky jako proof-of-work nebo proof-of-stake, které předcházejí případným snahám o ovlivnění blockchainu v zájmu útočníka.

Existují však i soukromé varianty blockchainu, které bývají označovány jako permissioned [48]. Pro přístup do takovéhoto systému je nutné získat autorizaci příslušné autority. Vzhledem k tomu, že tyto systémy vylučují neautorizované uživatele a dá se předpokládat určitá úroveň důvěry, nebývají zde využity techniky jako např. proof-of-work, ale výpočetně méně náročné způsoby dosažení shody při připojování nových bloků.

4.2 Smart kontrakt

Transakce probíhající na síti nemusejí mít pouze podobu převodu peněžních prostředků, ale mohou zahrnovat složitější instrukce a podmínky. Takovéto rozšíření možností blockchainu umožňují smart kontrakty.

Pojem smart kontrakt i jeho koncept navrhnul v roce 1997 Nick Szabo [41]. Smart kontrakty představuje jako způsob formalizace obchodních vztahů, který je bezpečnější a důvěryhodnější než tradiční „papírové“ dokumenty. Jako analogii uvádí prodejní automat, který při vložení dostatečné částky vydá zvolený produkt a vrátí rozdíl vkladu a ceny produktu.

Smart kontrakt představuje dohodu mezi dvěma a více stranami. Tradiční kontrakt, který by byl dokladem o takové dohodě, by obvykle vyžadoval dohled důvěryhodné třetí strany, aby bylo zajištěno splnění podmínek dohody všemi zúčastněnými stranami. Je-li dohoda uzavřena prostřednictvím smart kontraktu, na dodržení podmínek dohody dohlíží přímo sám smart kontrakt. Zúčastněné strany si tak nemusejí navzájem důvěřovat. Za důvěryhodnou třetí stranu lze považovat pouze blockchain, který uchovává a vykonává jejich smart kontrakt. Smart kontrakt je tvořen kódem a daty, je aktivován vložením prostředků pokrývajících náklady na vykonání kódu. Vykonání kódu smart kontraktu a uchování jeho dat zajišťují uzly sítě blockchainu.

Vlastnosti smart kontraktů vycházejí z principu fungování samotného blockchainu. Jsou decentralizované, jejich stav ani vykonávání není vázáno na jediný uzel sítě. Díky tomu také platí, že smart kontrakt je vykonán vždy tak, jak byl na blockchain vložen. Žádný z účastníků dohody ani uzly sítě nemají možnost ho pozměnit.

4.3 Platformy pro smart kontrakty

Ne každý blockchain je vhodný pro implementaci smart kontraktů. Například blockchain Bitcoinu, nejznámější kryptoměny využívající blockchain, je primárně určen pro provádění finančních transakcí a možnosti vytváření smart kontraktů na něm jsou omezené. Existují blockchainy, které se soustředí na možnost využití jako platformy pro smart kontrakty a jsou tak pro ně lépe přizpůsobeny.

Ethereum

Ethereum [17] je kryptoměnová platforma a zároveň virtuální stroj pro vytváření distribuovaných aplikací na blockchainu. Jednotka kryptoměny Etherea se nazývá ether, často bývá používána jednotka gwei, kde 1 gwei = 10^{-9} ether. Smart kontrakty na Ethereu jsou definovány pomocí nízkoúrovňového jazyka zvaného Ethereum Virtual Machine code (dále jako kód EVM), který je Turingovsky úplný. Nejčastěji jsou však smart kontrakty psány ve vysokoúrovňovém programovacím jazyce (jako např. Solidity nebo Vyper), který je do kódu EVM kompilován.

Každá instrukce v kódu EVM má definovanou cenu za provedení v množství jednotky zvané gas. Uzel sítě, který provede daný kód má nárok na odměnu (v etherech) odpovídající hodnotě gas vykonaných instrukcí. Cena jednotky gas v etherech se mění, v současné době (prosinec 2020) se pohybuje okolo 130 gwei za jednotku gas. Hodnota jednoho etheru je aktuálně přibližně 700 amerických dolarů [18].

Hlavními objekty tvořícími stav Etherea jsou účty. Těch jsou rozlišovány dva typy, externě vlastněné účty a účty kontraktů. Externě vlastněný účet je podobný klasickému bankovnímu účtu, má určitý zůstatek a vlastníka. Uživatel, který je vlastníkem účtu může nakládat s obnosem kryptoměny na účtu (provádět transakce). Účet kontraktu je ovládán kódem smart kontraktu, který je aktivován transakcí na daný účet. Smart kontrakt může být vytvořen uživatelem i jiným smart kontraktem. Každý účet je definován 20 bytovou adresou a obsahuje následující informace:

- hodnota nonce, což je čítač odeslaných transakcí z daného účtu (odlišná od hodnoty nonce bloku, která souvisí s obtížností těžby bloků u konceptu proof-of-work),
- zůstatek,
- úložiště dat,
- řídicí kód, jedná-li se o účet kontraktu.

Základním prostředkem komunikace je zpráva, sloužící k odeslání určitého množství kryptoměny a dat mezi účty. Zpráva může být odeslána jak z externě vlastněného účtu, tak z účtu kontraktu. Speciálním typem zprávy je transakce. Transakce je odesílána vždy z externě vlastněného účtu a kromě údajů o odesílateli, příjemci, množství odesílaných prostředků a dat obsahuje také hodnoty STARTGAS a GASPRICE. STARTGAS je maximální množství gas pro danou transakci. Tím limituje počet kroků výpočtu, které může daná transakce vyvolat, čímž je především předcházeno vzniku nekonečných smyček. GASPRICE je odměna (v množství etherů), kterou obdrží těžařský uzel sítě za jednotku gas. Je-li při provádění kódu překročen limit STARTGAS, je celý výpočet anulován, nicméně uzly, které jej provedly odměnu za výpočet obdrží. Pokud výpočet skončí, aniž by byl limit dosažen, je přebytek navrácen odesílateli transakce. Cílem je však co nejlépe odhadnout potřebný gas, jelikož je limitováno i celkové množství gas v jednom bloku (rozhodující je limit STARTGAS, nikoli skutečná spotřeba). Nadbytečně vysoký limit STARTGAS tedy snižuje pravděpodobnost, že bude transakce do bloku zařazena. Aktuální (prosinec 2020) limit pro jeden blok je 12 500 000 gas [18].

Qtum

Platforma Qtum [36] se snaží o propojení nejlepších vlastností Bitcoinu a Etherea. Základem Qtum je blockchain ve stejné podobě jako u Bitcoinu, tedy s modelem UTXO (Unspent Transaction Output).

Model UTXO popisuje způsob práce s prostředky, kdy vstupy nových transakcí jsou tvořeny výstupy dříve proběhlých transakcí. Zůstatek náležící konkrétní peněžence je tak dán prostředky ze všech transakcí, které byly poslány na adresu peněženky, pokud však tyto prostředky nebyly doposud použity jako vstupy odchozích transakcí (tedy utraceny). Tento model je analogií práce s fyzickými mincemi nebo bankovkami. Obdržíme-li např. minci v hodnotě 1 CZK v jeden den a poté další druhý den, můžeme následně utratit 2 CZK najednou, nebo opět po 1 CZK při dvou příležitostech, nemůžeme však ani jednu z mincí utratit vícekrát nebo ji rozdělit.

Pro realizaci smart kontraktů využívá Qtum princip virtuálního stroje (Ethereum Virtual Machine, zkráceně EVM) přejatý z Etherea. Ethereum však nevyužívá model UTXO, ale účty, které drží zůstatek prostředků. Pro propojení těchto dvou konceptů do jedné platformy zavádí Qtum vrstvu abstrakce účtu (Account Abstraction Layer), která mezi nimi vytváří rozhraní. Díky tomu je tato platforma kompatibilní se smart kontrakty napsanými pro Ethereum v jazyce Solidity. Vrstva abstrakce účtu ale není specifická pouze pro EVM a na Qtum tak může paralelně běžet několik virtuálních strojů. Kromě EVM nabízí Qtum i virtuální stroj x86, což vývojářům dává možnost využití mnoha existujících kompilátorů s minimálními úpravami. Dalším konceptem původem z Etherea, který Qtum využívá jsou poplatky v jednotce gas.

Na rozdíl od Bitcoinu i Etherea, které spoléhají na koncept proof-of-work, na Qtum je konsensu dosahováno pomocí varianty proof-of-stake, kde odměna za vytvoření bloku je rozdělena mezi deset posledních autorů bloků a vyplácena se zpožděním.

Qtum dále využívá smart kontraktů k realizaci protokolu pro řízení platformy (DGP). Tento protokol umožňuje provádět aktualizace parametrů (např. limitu velikosti bloku, ceny výpočetních operací a další) jednoduchým způsobem a zároveň dává uživatelům možnost podílet se hlasováním na rozhodování o tom, jaké změny budou provedeny. Díky DGP je rozdělení blockchainu (tzv. hard fork) nutné pouze v případě rozsáhlých změn.

NEO

NEO [30] je platforma pro stejnojmennou kryptoměnu a smart kontrakty, která vznikla v roce 2015 pod názvem Antshares. Měna NEO je nedělitelnou jednotkou a není již dále těžena. Celkem 100 milionů tokenů NEO bylo vytěženo najednou při spuštění sítě. Její cena se pohybuje okolo 15 amerických dolarů (prosinec 2020) [31]. Token gas je generován postupně při vzniku každého nového bloku a slouží jako poplatek za vykonání transakcí.

Většina běžných transakcí probíhá zdarma, poplatek je nutný pro složitější transakce, které přesáhnou velikost 1024 bytů (povolená velikost transakce je až 100 kB). U menších transakcí slouží poplatek ke zvýšení priority transakce. Počet bezplatných transakcí v jednom bloku je omezen na 20, a zaplacení poplatku tak může urychlit zařazení transakce do bloku, především při velkém zatížení sítě. Celkem může být v jednom bloku maximálně 500 transakcí.

Pro dosažení konsensu využívá protokol dBFT (delegated Byzantine fault-tolerant). Tohoto protokolu se účastní pouze několik vybraných uzlů, které generují nové bloky.

Smart kontrakty na platformě NEO lze psát v několika programovacích jazycích jako např. C#, C++ nebo Java. Aktivace smart kontraktu může proběhnout pomocí transakce odeslané uživatelem nebo voláním z jiného smart kontraktu (nepovoluje rekurzivní volání). Za zařazení smart kontraktu do blockchainu i za jeho vykonání může být účtován poplatek. Pro zvýšení zájmu uživatelů jsou poplatky minimalizovány, ačkoli je tedy fixní poplatek za většinu operací stanoven na 0.001 gas, prvních 10 gas není účtováno.

Cardano

Cílem platformy Cardano [7] je vytvoření spolehlivého a bezpečného permissionless blockchainu, její vývoj probíhá ve spolupráci s odborníky v oblasti kryptografie. Implementačním jazykem platformy je funkcionální programovací jazyk Haskell a celá implementace je dostupná jako open-source. Cardano využívá proof-of-stake protokol zvaný Ouroboros a vlastní kryptoměnu Ada.

Platforma Cardano je stále předmětem vývoje a některé její funkcionality nejsou ještě implementovány nebo připraveny k použití. To se týká i smart kontraktů, ty by se na Cardanu měly objevit v rámci třetí vývojové fáze zvané Goguen v průběhu roku 2021. Smart kontrakty na Cardanu budou vytvářeny v jazyce Plutus. Ten je navržen přímo pro vytváření smart kontraktů, jeho základem je Haskell. Právě využití funkcionálního programovacího jazyka by mělo umožnit pokročilé testování a verifikaci smart kontraktů na platformě Cardano.

Hyperledger Fabric

Hyperledger Fabric [2], součást projektu Hyperledger konsorcia Linux Foundation, je framework poskytující nástroje pro vytváření vlastního permissioned blockchainu. Díky modulárnímu přístupu umožňuje modifikovat vlastnosti výsledného systému dle potřeb konkrétního cílového využití.

Smart kontrakty na platformě Fabric jsou součástí tzv. chaincode. Pojmy smart kontrakt a chaincode jsou často v kontextu této platformy používány jako synonyma, přesněji je však chaincode množinou více smart kontraktů, které definují operace nad jednou konkrétní entitou.

Součástí sítě Fabric je několik typů uzlů, které plní různé úlohy, jako provádění chaincode a transakcí nebo zajišťování konzistence blockchainu. Implementace způsobu dosažení shody o pořadí transakcí a podobě blockchainu je jednou z modulárních komponent a je tedy možné zvolit řešení vhodné pro konkrétní použití. Zvláštním prvkem sítě jsou uzly MSP (Membership Service Providers), které slouží k ověřování identit a oprávnění členů sítě.

Jednou z výhod platformy Fabric je možnost psát chaincode v běžně používaných programovacích jazycích (např. Go, Java). Součástí platformy Fabric není žádná kryptoměna.

EOS

EOS [16] je open-source blockchain a platforma pro smart kontrakty zahrnující stejnojmennou kryptoměnu. Využívá konceptu delegovaného proof-of-stake (DPOS), kde držitelé tokenů hlasováním vybírají uzly, kterým bude umožněno vytvářet nové bloky. Bloky jsou přidávány každé půl sekundy v kolech po 126 blocích. Pro každé kolo je vybráno vždy 21 uzlů, z nichž každý má možnost vytvořit až 6 bloků. Aktuálně zvolené uzly se střídají v předem dohodnutém pořadí v přidávání bloků a po ukončení jejich kola jsou nahrazeny další skupinou zvolených uzlů. Za provedené transakce nejsou účtovány poplatky, uzly vytvářející bloky jsou odměňovány nově vytvářenými tokeny. Celkové množství kryptoměny EOS každoročně narůstá o 5 %.

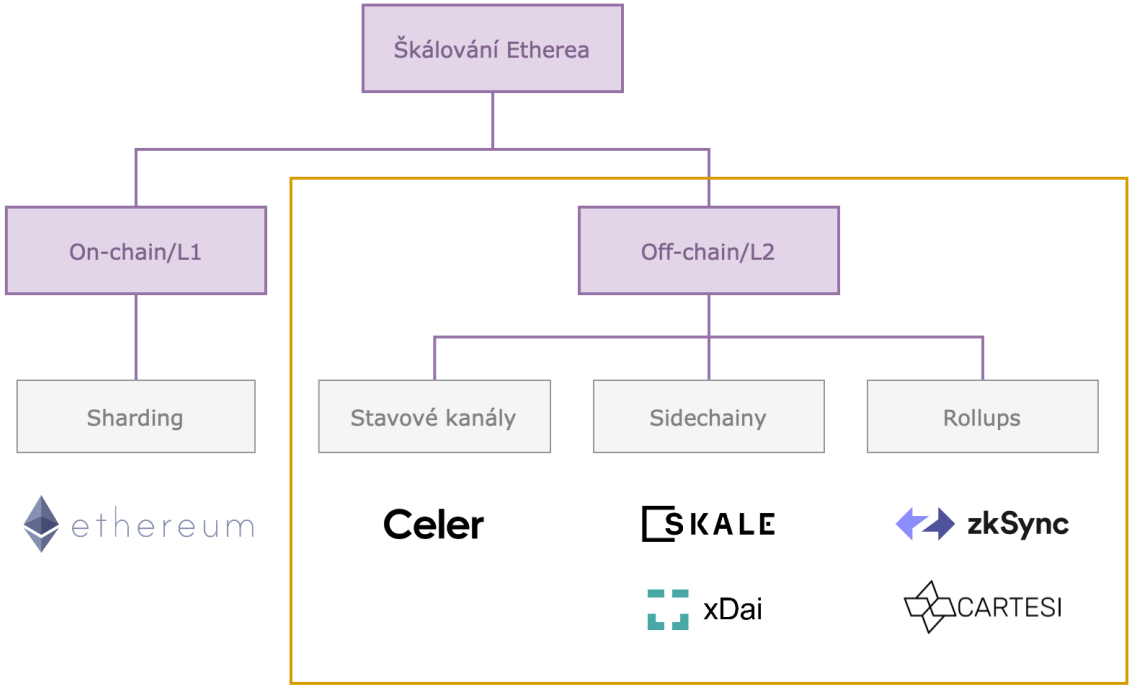
Každá transakce na blockchainu EOS musí zahrnovat část hodnoty hash aktuálního bloku. Tím jsou omezeny možné útoky zneužívající rozdělení blockchainu na několik větví. Transakce může být provedena vždy na jedné větvi blockchainu, na jiné by v ní obsažená hodnota hash neodpovídala.

Blockchain EOS je typu permissioned. Vytvoření nového uživatelského účtu vyžaduje potvrzení již registrovaného uživatele a zaplacení poplatku za paměťový prostor.

Ačkoli transakce na EOS zpoplatněné nejsou, vytvoření smart kontraktu i běh jeho výpočtů vyžaduje zaplacení poplatků jak za paměťové úložiště označované jako RAM, tak za zdroje nutné pro výpočet (čas CPU, šířka pásma).

4.4 Nadstavbová řešení

Jedním z problémů, kterým čelí blockchainové platformy jako Bitcoin nebo Ethereum je škálovatelnost. Zvýšení počtu aktivních uživatelů s rostoucí popularitou blockchainu vede k zahlcení sítě, dlouhému čekání na potvrzení transakcí a vysokým cenám za provedené transakce. U nově vznikajících platforem může být tento problém uvažován již od počátku, přičemž příkladem možného řešení je využití lépe škálovatelných protokolů pro dosažení konsensu. U již existujících blockchainů je možné podobná vylepšení zavést po provedení hard-forku, nebo škálovatelnost řešit vytvořením nadstavby nad existující platformou



Obrázek 4.2: Přehled možností škálování platformy Ethereum s příklady zástupců daných přístupů [34].

(angl. second-layer nebo layer 2). Pro Ethereum jsou plánovány změny zlepšující výkonnost sítě přímo na první vrstvě. Platforma přejde na protokol proof-of-stake a zavede tzv. sharding, rozdělení blockchainu na několik paralelních, přičemž každý z nich bude dosahovat stejných výkonových parametrů jako současný jediný hlavní řetězec.

Projekty tvořící druhou vrstvu nevyžadují změny ve fungování blockchainu (na první vrstvě), ale umožňují zpracování transakcí mimo tento řetězec, čímž snižují zátěž hlavní sítě. Konkrétní způsob fungování se liší u jednotlivých projektů. Nadstavby nad Ethereem využívají nejčastěji jeden ze tří přístupů nebo jejich kombinaci [34], jak je znázorněno na obrázku 4.2.

Prvním ze tří zmíněných přístupů jsou stavové kanály (angl. state channels). Ty umožňují provedení většího množství transakcí mezi skupinou uživatelů s nutností zaznamenání pouze dvou transakcí na blockchainu. První transakcí jsou pro tento účel vyhrazeny potřebné prostředky (otevření stavového kanálu) a následné nakládání s těmito prostředky probíhá bez interakce s řetězcem. Na ten je pouze v druhé transakci zaznamenán výsledný stav (uzavření kanálu).

Druhým možným přístupem je tzv. roll-up, řešení, které provádí transakce mimo blockchain, na něj zaznamenává pouze validační kryptografické důkazy pro velké množství transakcí najednou.

Dalším nadstavbovým řešením jsou sidechainy, oddělené a nezávisle fungující řetězce. Tyto řetězce mají vlastní validační uzly, protokol pro dosažení konsensu i tokeny, umožňují však převedení prostředků z blockchainu, který rozšiřují.

4.4.1 xDai

Blockchain xDai [47] je zástupcem nadstavbového řešení nad Ethereem formou sidechainu. Využívá konceptu proof-of-stake, pro který je zde vyžadován token STAKE – jeden ze dvou tokenů na tomto blockchainu. Druhý token, kryptoměna xDai, patří mezi tzv. stablecoiny, kryptoměny jejichž hodnota je neměnná vůči jiné měně. V tomto případě, hodnota jednoho xDai odpovídá přibližně jednomu americkému dolaru. To zajišťuje stabilní, předvídatelné ceny transakcí, jelikož právě token xDai je využíván i na transakční poplatky na xDai blockchainu.

Mezi hlavní výhody platformy xDai patří především rychlé zpracování transakcí (nový blok je vytvořen každých 5 sekund), nízké ceny transakcí (cena jednotky gas je trvale 1 gwei) a kompatibilita s Ethereem. Nevýhodou může být naopak menší úroveň decentralizace. Vzhledem k závislosti na Ethereu, blockchain xDai s Ethereem sdílí shodné parametry jako jsou ceny instrukcí v gas nebo limit velikosti bloku. Také však umožňuje vývojářům převedení jejich aplikací z Etherea bez nutnosti rozsáhlých změn. Tím mohou dosáhnout výrazného snížení nákladů oproti provozování aplikace na Ethereu.

Tokeny xDai je možné získat převedením kryptoměny DAI z Etherea. Pro tento účel jsou vytvořena přemostění mezi blockchainem Etherea a xDai (OmniBridge a xDai Bridge).

Kapitola 5

Návrh škálovatelného hlasovacího systému

5.1 Zero-knowledge důkazy

Zero-knowledge důkaz [26] slouží k prokázání znalosti určité utajené hodnoty, bez prozrazení jakékoli informace o této hodnotě. Pro provedení důkazu musí spolupracovat dva účastníci. Jeden, který svou znalost určité hodnoty dokazuje a druhý, který ji ověřuje.

Interaktivní

Interaktivní zero-knowledge důkaz obecně probíhá tak, že ověřující strana předá dokazující straně výzvu (challenge), která má podobu náhodně zvolené hodnoty. Na tuto výzvu může dokazující strana poskytnout očekávanou odpověď pouze pokud příslušnou hodnotu skutečně zná.

Neinteraktivní

Důkaz může proběhnout i bez nutnosti interakce mezi oběma stranami. Princip důkazu zůstává stejný, pouze nedochází k předání výzvy od ověřující strany. Hodnota výzvy je vytvořena samotnou dokazující stranou pomocí hashovací funkce.

Konkrétní důkazy využité v návrhu hlasovacího systému v této práci jsou neinteraktivní a ověřující stranu pro ně představuje smart kontrakt. Prvním z těchto důkazů (viz obrázek 5.1) každý hlasující ukazuje korektnost svého hlasovacího lístku. Podstatou zde je dokázání náležitosti hlasujícím zvolené hodnoty (hlasu) do dané množiny (možných kandidátů). Druhý důkaz (viz obrázek 5.2) je využit v opravné fázi protokolu a dokazuje, že hlasujícím P_i poskytnutá hodnota $g^{x_i x_j}$ odpovídá jeho vlastnímu klíči g^{x_i} a klíči g^{x_j} hlasujícího P_j.

5.2 Protokol hlasování

Navržený protokol využívá smart kontraktů na blockchainu. Díky homomorfnímu šifrování hlasovacích lístků protokol zajišťuje ochranu soukromí hlasujících. Dále má vlastnost self-tallying, je verifikovatelný a odolný proti selhání v důsledku neúčasti několika registrovaných hlasujících. Návrh vychází především z protokolů od Hao et al. [24] a BBB-voting [42] a pro

Hlasující P_i		*Smart kontrakt*
$h \leftarrow g^{y_i}$		$h \leftarrow g^{y_i}$
Zvolí kandidáta $v_i \in \{1, ..., k\}$		
$f_l \in \{f_1, ..., f_k\} \subseteq \mathbb{F}_p^*$		
Zveřejní:		
$x \leftarrow g^{x_i}$		
$B_i \leftarrow h^{x_i} f_l$		
$w \in_R \mathbb{F}_p^*$		
$\forall l \in \{1, ..., k\} \setminus v_i$:		
1. $r_l, d_l \in_R \mathbb{F}_p^*$		
2. $a_l \leftarrow x^{-d_l} g^{r_l}$		
3. $b_l \leftarrow h^{r_l} (\frac{B_i}{f_l})^{-d_l}$		
for v_i:		
1. $a_{v_i} \leftarrow g^w$		
2. $b_{v_i} \leftarrow h^w$		
$c \leftarrow H_{zk}(\{\{a_l\}, \{b_l\}\}_l)$		
for v_i:		
1. $d_{v_i} \leftarrow \sum_{l \neq v_i} d_l$		
2. $d_{v_i} \leftarrow c - d_{v_i}$		
3. $r_{v_i} \leftarrow w + x_i d_{v_i}$		
4. $q \leftarrow p - 1$		
5. $r_{v_i} \leftarrow r_{v_i} \mod q$		
	$\xrightarrow{\forall l: \{a_l\}, \{b_l\}, \{r_l\}, \{d_l\}}$	
		$\Psi \leftarrow \{\forall l : \{a_l\}, \{b_l\}\}$
		$c \leftarrow H_{zk}(\Psi)$
		$\sum_l d_l \stackrel{?}{=} c$
		$\forall l \in \{1, ..., k\}$:
		1. $g^{r_l} \stackrel{?}{=} a_l x^{d_l}$
		2. $h^{r_l} \stackrel{?}{=} b_l (\frac{B_i}{f_l})^{-d_l}$

Obrázek 5.1: Neinteraktivní zero-knowledge důkaz korektnosti hlasovacího lístku. [42]

Hlasující P_i		*Smart kontrakt*
$A \leftarrow g^{x_i}$, $B \leftarrow g^{x_j}$, x_i		$A \leftarrow g^{x_i}$, $B \leftarrow g^{x_j}$
$w_i \in_r \mathbb{F}_p$		
$C \leftarrow g^{x_i x_j}$		
$m_1 \leftarrow g^{w_i}$		
$m_2 \leftarrow B^{w_i}$		
$c \leftarrow H_{zk}(A, B, m_1, m_2)$		
$r_i \leftarrow w_i + c x_i$		
	$\xrightarrow{C, r_i, m_1, m_2}$	$c \leftarrow H_{zk}(A, B, m_1, m_2)$
		$g^r \stackrel{?}{=} m_1 A^c$
		$B^r \stackrel{?}{=} m_2 C^c$

Obrázek 5.2: Neinteraktivní zero-knowledge důkaz korektnosti hodnoty $g^{x_i x_j}$ pro opravný krok hlasování. [42]

umožnění hlasování více kandidátů využívá princip z protokolu dle Kiayias a Yung [26]. Na rozdíl od těchto protokolů je škálovatelný.

Účastníci protokolu

Autorita spravující hlasování Úkolem autority je správa průběhu hlasování. Její přítomnost nemá vliv na vlastnosti protokolu, tzn. autorita nemá možnost odhalit obsah hlasovacích lístků ani ovlivnit výsledek. Autorita také definuje časové intervaly pro jednotlivé fáze protokolu.

Hlasující Seznam oprávněných hlasujících je spravován autoritou. Hlasující musí pro účast v hlasování registrovat svůj kryptografický klíč a následně odevzdat platný hlasovací lístek.

Blockchainová platforma Blockchainová platforma slouží v protokolu jako vývěska pro zveřejnění údajů v průběhu hlasování a také jako výpočetní platforma, stejně jako je tomu v podobných protokolech, např. Open Vote Network [28].

Pro kontrolu korektnosti průběhu hlasování je možná účast i další strany, např. auditora. Veškeré údaje o průběhu hlasování jsou veřejně dostupné na blockchainu.

Průběh hlasování

Autorita musí nejprve ověřit identitu každého hlasujícího a zajistit, že každému hlasujícímu odpovídá jediná adresa peněženky. Jakým způsobem je toto realizováno není předmětem této práce. Samotný protokol pak probíhá v těchto krocích:

1. krok: Rozdělení hlasujících do skupin

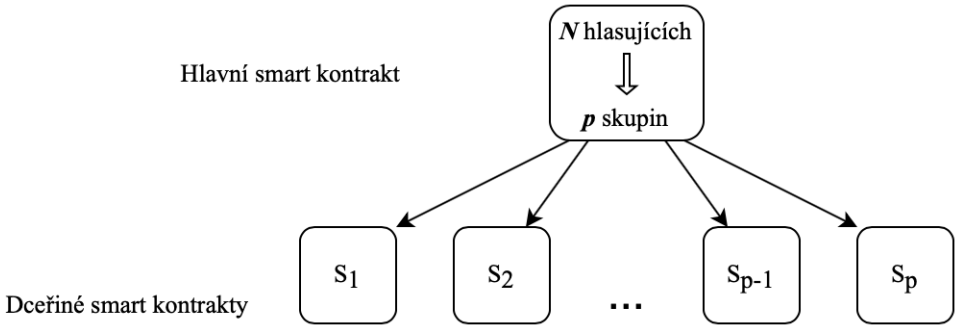
Autorita odešle adresy peněženek všech hlasujících na účet hlavního smart kontraktu v tolika transakcích, kolik je potřeba, aby při daném počtu hlasujících nebyl překročen limit velikosti bloku. Hlavní smart kontrakt následně náhodně rozdělí hlasující do skupin o takové velikosti, jakou umožňuje použitá implementační platforma s ohledem na náročnost výpočtu výsledku hlasování. Pro každou skupinu hlasujících je hlavním smart kontraktem vytvořen samostatný smart kontrakt (viz obrázek 5.3). Tento dceřiný smart kontrakt realizuje následující kroky pouze v rámci do něj spadající skupiny hlasujících.

2. krok: Parametry hlasování a registrace klíčů

Pro každou ze skupin hlasujících jsou určeny parametry hlasování. Připadá-li do skupiny n hlasujících a je-li možnost volby mezi k kandidáty, jsou parametry následující:

- generátor $g \in \mathbb{F}_p^*$, kde $p = 2q + 1$, q je prvočíslo a $n < p - 1$,
- k nezávislých generátorů $\{f_1, ..., f_k\}$ z $\mathbb{F}_p^*$ splňujících podmínku [24]: $f_i = g^{2^{(i-1)m}}$, kde $i \in 1...k$ a m je nejmenší celé číslo takové, že $2^m > n$.

Každý hlasující P_i vytvoří své klíče, tajný a veřejný. Tajným klíčem je náhodná hodnota $x_i \in_R \mathbb{F}_p^*$ a veřejným klíčem je hodnota g^{x_i}. Veřejné klíče jsou zaslány hlasujícími na účet smart kontaktu.



Obrázek 5.3: Znázornění hlavního smart kontraktu vytvářejícího dceřiné smart kontrakty pro skupiny hlasujících.

3. krok: Synchronizace klíčů

V rámci každé ze skupin jsou veřejné hlasovací klíče použity k výpočtu hlasovacích klíčů pro jednotlivé hlasující. Nejprve je smart kontraktem vypočítána sdílená část hlasovacího klíče g^{y_i} pro každého hlasujícího P_i:

$$g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} / \prod_{j=i+1}^{n} g^{x_j} \tag{5.1}$$

Zde platí (ukázáno v [24]), že $\sum_i x_i y_i = 0$. Samotný hlasovací klíč každého hlasujícího P_i je pak $g^{x_i y_i}$. Jelikož x_i je tajný klíč, pouze hlasující P_i je schopný vypočítat svůj hlasovací klíč.

4. krok: Hlasování

Pro vytvoření hlasovacího lístku potřebuje hlasující svůj hlasovací klíč a generátor kandidáta, kterému chce dát svůj hlas. Hlasovací klíč zajistí utajení tohoto hlasu. Je proto ale nutné, aby hlasující dokázal, že hlas na jeho lístku je korektní a odpovídá parametrům hlasování.

Hlasovací lístek má podobu $B_i = g^{x_i y_i} f_j$, kde $f_j \in f_1, ..., f_k$ určuje vybraného kandidáta. Tento lístek hlasující zveřejní tím, že jej odešle smart kontraktu příslušné skupiny. Spolu s tímto lístkem odešle každý hlasující také neinteraktivní zero-knowledge důkaz dle algoritmu 5.1, který je ověřen smart kontraktem.

5. krok: Vyloučení neaktivních hlasujících

Pokud někteří z hlasujících, kteří se zúčastnili prvních dvou kroků hlasování, neodevzdají ve třetím kroku svůj hlasovací lístek, není možné přejít k výpočtu výsledku. Sdílení části hlasovacích klíčů mezi všemi hlasujícími je příčinnou tohoto problému, zároveň ale umožňuje části klíčů neaktivních hlasujících s platných hlasů odstranit. Tato fáze opět probíhá zvlášť v každé skupině hlasujících.

Jako příklad můžeme uvažovat situaci, kdy se hlasování účastní čtyři hlasující P_1, P_2, P_3 a P_4, ale P_4 neodevzdá hlas ve třetím kroku. Každý z hlasujících P_1, P_2 a P_3 dokáže vypočítat část svého hlasovacího klíče, kterou sdílí s neaktivním hlasujícím P_4. Potřebný údaj od P_4, jeho veřejný klíč g^{x_4}, je všem znám, zároveň každý zná svůj vlastní tajný klíč. Hlasující P_1 tedy vypočítá $g^{x_1 x_4}$, stejně postupují P_2 a P_3 se svými tajnými klíči a získají

$g^{x_2x_4}$ a $g^{x_3x_4}$. Tyto hodnoty jsou odeslány smart kontraktu spolu z důkazem (viz 5.2), že odeslaná hodnota skutečně odpovídá patřičným klíčům.

Oprava korektně odevzdaných hlasovacích lístků B_1, B_2, B_3 je znázorněna na obrázku 5.4.

Odevzdané hlasovací lístky	
$B_1 = g^{-x_1x_2} \cdot g^{-x_1x_3} \cdot g^{-x_1\mathbf{x_4}} \cdot f_1$	
$B_2 = g^{x_1x_2} \cdot g^{-x_2x_3} \cdot g^{-x_2\mathbf{x_4}} \cdot f_2$	
$B_3 = g^{x_1x_3} \cdot g^{x_2x_3} \cdot g^{-x_3\mathbf{x_4}} \cdot f_3$	
Opravný výpočet	*Výsledné hlasovací lístky*
$B_1 = g^{-x_1x_2} \cdot g^{-x_1x_3} \cdot g^{-\mathbf{x_1x_4}} \cdot g^{\mathbf{x_1x_4}} \cdot f_1$	$= g^{-x_1x_2} \cdot g^{-x_1x_3} \cdot f_1$
$B_2 = g^{x_1x_2} \cdot g^{-x_2x_3} \cdot g^{-\mathbf{x_2x_4}} \cdot g^{\mathbf{x_2x_4}} \cdot f_2$	$= g^{x_1x_2} \cdot g^{-x_2x_3} \cdot f_2$
$B_3 = g^{x_1x_3} \cdot g^{x_2x_3} \cdot g^{-\mathbf{x_3x_4}} \cdot g^{\mathbf{x_3x_4}} \cdot f_3$	$= g^{x_1x_3} \cdot g^{x_2x_3} \cdot f_3$

Obrázek 5.4: Oprava lístků hlasujících P_1, P_2 a P_3 při vyloučení neaktivního hlasujícího P_4.

Tato oprava může být provedena pro několik neaktivních účastníků zároveň. Pokud se některý z hlasujících nezúčastní opravného kroku, je možné tento krok opakovat pro dodatečné vyloučení i těchto hlasujících.

6. krok: Výpočet výsledků ve skupinách

Výsledné počty hlasů c_i, $\forall i \in \{1, ..., k\}$ pro jednotlivé kandidáty v rámci každé skupiny hlasujících jsou vypočítány nalezením řešení rovnice

$$\prod_{i=1}^{n} B_i = \prod_{i=1}^{n} g^{x_i y_i} f = g^{\sum_i x_i y_i} f = f_1^{c_1} f_2^{c_2} ... f_k^{c_k} \tag{5.2}$$

Pro nalezení řešení této rovnice je nutné využít prohledávání všech možných hodnot (exhaustive search). Kvůli výpočetní náročnosti je tento výpočet prováděn mimo smart kontrakt kterýmkoli účastníkem hlasovacího protokolu (autoritou nebo hlasujícími, možný je i distribuovaný výpočet nebo využití výkonnější výpočetní platformy, např. cloudu). Úkolem smart kontraktu je pak pouze ověřit platnost rovnice 5.2.

7. krok: Agregace výsledku

Poté co v každé ze skupin hlasujících dojde k ukončení hlasování (včetně opravných kroků a ověření výsledku), je nutné výsledky jednotlivých skupin spojit do celkového výsledku hlasování. To provede hlavní smart kontrakt agregací výsledků ze všech dceřiných smart kontraktů.

Kapitola 6

Implementace návrhu

Tato kapitola je věnována výběru vhodné platformy a podobě samotné implementace navrženého protokolu. Popsána bude struktura implementace, i jakým způsobem implementace realizuje průběh hlasování. Vysvětleny budou také techniky vedoucí k optimalizaci potřebných výpočtů.

6.1 Zvolená platforma

Pro vytvoření implementace demonstrující funkčnost navrženého hlasovacího protokolu byla původně uvažována platforma Neo vzhledem k vlastnostem jako je rychlost generování nových bloků, cena vytváření a provozování smart kontraktů nebo aktuální aktivita vývoje této platformy. Na blockchainu Neo je nový blok generován přibližně každých 15 sekund, což je srovnatelná hodnota s průměrnými 13 sekundami na Ethereu [18] a lepší ve srovnání s 2 minutami v případě platformy Qtum. EOS blockchain je podstatně rychlejší, nový blok je generován v 0,5 sekundových intervalech, nicméně cena provozování smart kontraktů závisí na výši poplatků za omezené zdroje jako paměťové úložiště a procesorový čas, a dá se tak předpokládat její růst vzhledem ke zvyšujícímu se zájmu o tuto platformu. Neo také nabízí více možností výběru implementačního jazyka a vývojových nástrojů.

Nevýhodou platformy Neo se ukázal být aktuální stav vývoje této platformy. Starší verze Neo2 se již nadále nerozvíjí, je pouze minimálně udržována, a bude nahrazena novou verzí Neo3. Ta je však stále předmětem velmi aktivního vývoje, čemuž odpovídá i stav dostupných vývojových nástrojů. Dalším a zároveň nejpodstatnějším problémem se ukázala být nemožnost vytváření nových (dceřiných) smart kontraktů již existujícím kontraktem. To je dáno především tím, že Neo využívá jako unikátní identifikátor kontraktu hash kódu kontraktu a adresy odesílatele transakce vytvářející kontrakt. Není tedy možné odlišit více klonů jednoho kontraktu vytvořených jediným odesílatelem. Vytvoření dceřiných kontraktů jednoho hlavního kontraktu je koncept, na kterém závisí škálovatelnost vytvořeného návrhu, platformu Neo tak pro jeho implementaci nebylo možné využít.

Implementaci konceptu dceřiných kontraktů umožňuje programovací jazyk Solidity, který je přímo určený k vytváření kontraktů pro Ethereum Virtual Machine (EVM). Alternativní platformy pro smart kontrakty (jako např. právě Neo) využívají obecné programovací jazyky (tedy nikoli určené výhradně pro implementaci smart kontraktů), ačkoli podporují pouze podmnožinu jejich funkcionalit. Implementace kontraktů v jazyce Solidity je možná na platformě Qtum, která pro smart kontrakty využívá EVM, zajímavějšími vlastnostmi však disponuje platforma xDai. Jelikož se jedná o nadstavbové řešení nad Ethereem, je

plně kompatibilní s pro něj vytvořenými smart kontrakty, čímž umožňuje například přenesení stávajících kontraktů, které jsou na Ethereu příliš nákladné. Protože právě snížení nákladů je hlavním důvodem pro využití alternativního blockchainu a nikoli Etherea, byl pro realizaci hlasovacího protokolu zvolen právě xDai s implementací kontraktů v jazyce Solidity.

6.2 Struktura implementace

Jelikož návrh hlasovacího protokolu vychází primárně z protokolu BBB-voting [42], byla také implementace BBB-voting využita jako základ pro implementaci protokolu navrženého v této práci. BBB-voting je tvořen jedním smart kontraktem, se kterým komunikuje autorita spravující hlasování i samotní voliči, a dále knihovnami pro kryptografické operace. Pro dosažení lepší škálovatelnosti bylo nutné tuto implementaci upravit tak, aby bylo možné využít více smart kontraktů, mezi které by jeden hlavní kontrakt rozdělil registrované voliče podobně, jako jsou v klasických volbách voliči rozděleni mezi volební okrsky nebo místnosti. V tomto modelu pak každý volič komunikuje pouze s dceřiným kontraktem, do jehož skupiny byl přidělen. S hlavním kontraktem komunikuje pouze volební autorita za účelem zaregistrování hlasujících a dalších úkonů spojených ze zajištěním průběhu hlasování.

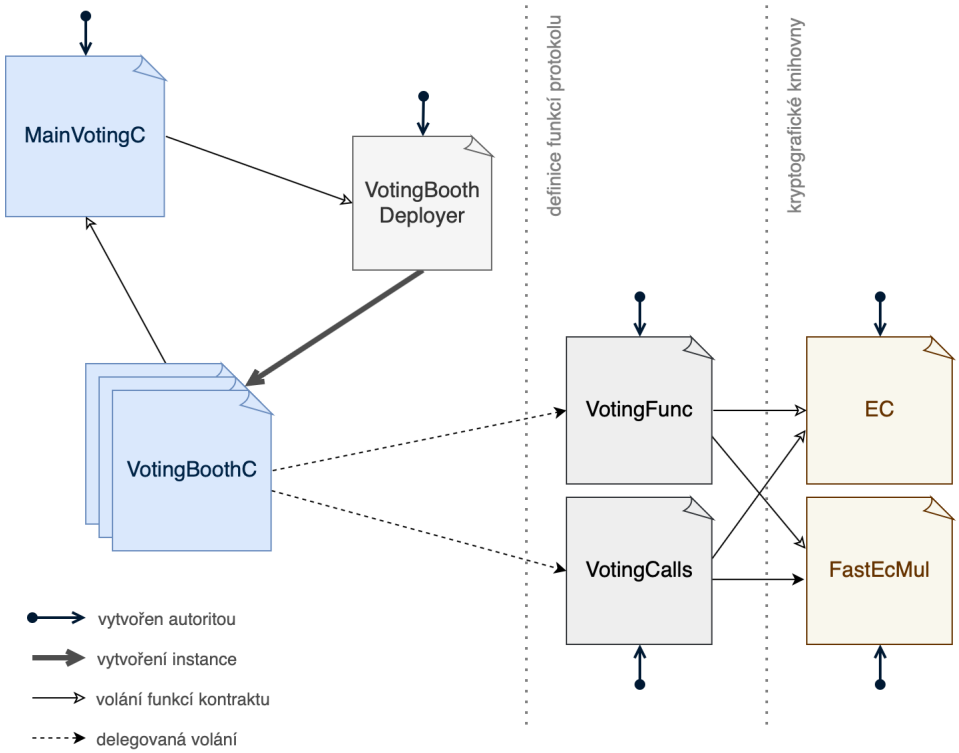
Kontrakty hlasovacího systému

Implementace hlasovacího systému je tvořena celkem pěti smart kontrakty a dvěma knihovnami pro kryptografické operace, vše v jazyce Solidity. Implementaci bylo nutné rozdělit na více částí, než pouze na hlavní agregační kontrakt a kontrakt pro hlasování skupin, jelikož skupinový kontrakt by tak při vytvoření překračoval limit velikosti bloku. Výslednou strukturu implementace a cesty komunikace mezi jednotlivými kontrakty ilustruje schéma na obrázku 6.1. Role jednotlivých kontraktů jsou popsány dále.

Hlavní kontrakt Hlavní kontrakt, v implementaci nazván `MainVotingC` vykonává úkony nutné k zastřešení a agregaci hlasování v jednotlivých skupinách.

Kontrakt skupiny hlasujících Průběh hlasování v jednotlivých skupinách spravuje kontrakt `VotingBoothC`. Nová instance tohoto kontraktu je vytvořena pro každou hlasovací skupinu, čímž je zajištěno, že v každé této skupině probíhá shodný hlasovací protokol jako ve všech ostatních. Vzhledem k tomu, že může být nutné vytvoření velkého počtu kopií tohoto kontraktu, je s ohledem na finanční náklady žádoucí, aby byl tento kontrakt co nejmenší a jeho vytvoření tak co nejlevnější. Proto je `VotingBoothC` jen jakousi kostrou hlasovacího protokolu, implementace všech funkcí je přesunuta do dalších dvou kontraktů. Toto oddělení nejen snižuje náklady na vícenásobné vytváření kontraktu, bylo také nutné proto, aby tento kontrakt bylo možné nasadit při aktuálním limitu velikosti bloku platformy xDai.

Definice funkcí protokolu Implementace funkcí z `VotingBoothC` je přesunuta do kontraktů `VotingFunc` a `VotingCalls`. První z nich definuje funkce, které zasahují do úložiště kontraktu – jsou vykonány tzv. on-chain, jejich volání vyžaduje vytvoření transakce na blockchainu a je tedy zpoplatněno. Druhý z těchto kontraktů pak definuje funkce, které mohou být volány tzv. off-chain. Nemodifikují úložiště kontraktu a mohou být vykonány



Obrázek 6.1: Znázornění struktury hlasovacího systému a interakcí mezi jednotlivými kontrakty.

pouze na lokálním uzlu bez nutnosti zaznamenání transakce na blockchain. Oba tyto kontrakty jsou vytvořeny jen jednou, bez ohledu na počet instancí kontraktu `VotingBoothC`.

Jednotlivé instance `VotingBoothC` pak přistupují k funkcím v kontraktech `VotingFunc` a `VotingCalls` pomocí delegování volání funkce `delegatecall`. Delegování umožní provedení kódu cílového kontraktu v kontextu delegujícího kontraktu. To znamená, že delegovaná funkce nepracuje s úložištěm kontraktu, který tuto funkci definuje, ale přistupuje k úložišti kontraktu, který funkci zavolal pomocí `delegatecall`. Díky tomu tak funkce z kontraktů `VotingFunc` a `VotingCalls` pracují z údaji hlasovací skupiny z instance `VotingBoothC`, aniž by tyto údaje bylo nutné předávat jako funkční parametry a případné změny přebírat s návratových hodnot funkcí. Vzhledem k počtu proměnných, se kterými je nutné v těchto funkcích pracovat, by takový přístup vedl k problémům s nedostatkem místa na zásobníku. Nevýhodou použití `delegatecall` je nutnost zachování stejné struktury úložiště u delegujícího a cílového kontraktu. Kontakty `VotingFunc` a `VotingCalls` musí definovat ve svém úložišti stejné proměnné ve stejném pořadí jako `VotingBoothC`, aniž by přímo se svým úložištěm pracovaly, což se projeví v ceně vytvoření těchto kontraktů.

Vytváření kontraktů skupin Aby mohl hlavní kontrakt `MainVotingC` vytvářet dceřiné kontrakty pro jednotlivé hlasovací skupiny, musí mít k dispozici kód kontraktu skupin `VotingBoothC`. Vložení kódu `VotingBoothC` přímo do `MainVotingC` pomocí `import` však opět vede na kontrakt, který je příliš velký pro vytvoření jedinou transakcí. Tento problém řeší kontrakt `VotingBoothDeployer`, který má jediný úkol – vytvářet nové instance `VotingBoothC` na žádost hlavního kontraktu.

Knihovny Knihovny `EC` a `FastEcMul` pro kryptografické operace nad eliptickými křivkami jsou převzaty z The Witnet Project [45] a doplněny o funkce využité v Open Vote Network [28] a BBB-voting [42].

6.3 Průběh hlasování

Předpokladem pro zahájení hlasování je nasazení všech potřebných smart kontraktů, kterými jsou hlavní kontrakt `MainVotingC`, kontrakty s funkčními definicemi `VotingFunc` a `VotingCalls`, obě kryptografické knihovny a pomocný kontrakt pro vytváření dceřiných kontraktů `VotingBoothDeployer`. Tento krok provádí autorita spravující hlasování. Konstruktoru hlavního kontraktu `MainVotingC` jsou již při vytváření předány údaje o kandidátech, tedy jejich popis a kryptografické generátory. Jsou-li tyto kontrakty připraveny na blockchainu, může být přistoupeno k jednotlivým krokům hlasovacího protokolu. Hlasovací protokol je rozdělen do šesti fází – `SETUP`, `SIGNUP`, `PRE_VOTING`, `VOTING`, `FAULT_REPAIR` a `TALLY`. Konkrétní úkony spadající do těchto fází budou dále podrobněji popsány.

Kontrakt `MainVotingC` a jednotlivé instance `VotingBoothC` udržují informaci o aktuální fázi a při volání funkcí těchto kontraktů (ať autoritou nebo hlasujícími) je ověřováno, zdali se hlasování nachází ve fázi, pro kterou je daná funkce relevantní. Vymezení časových intervalů pro jednotlivé fáze hlasování není pro jednoduchost součástí implementace.

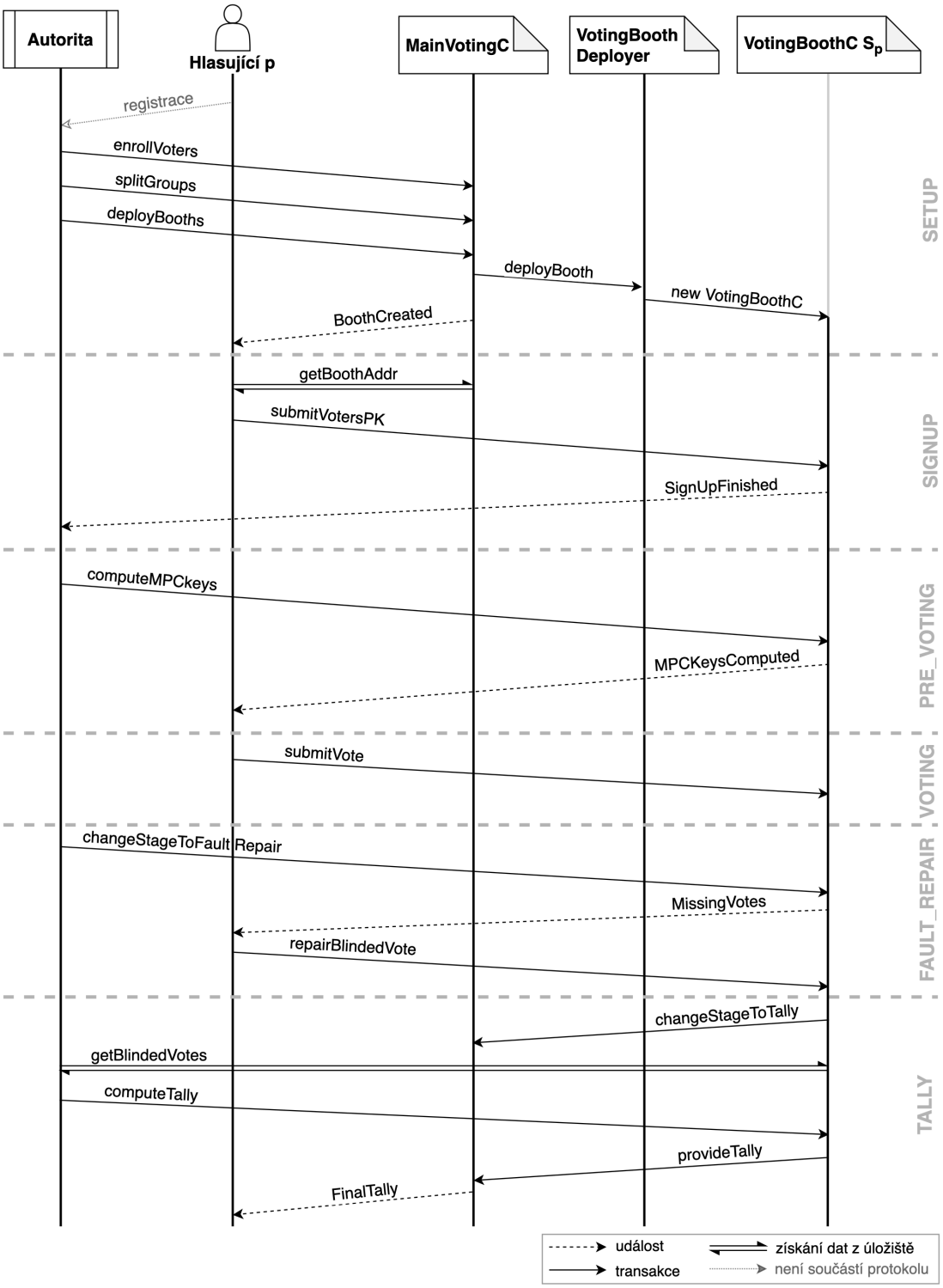
Obrázek 6.2 ukazuje schéma průběhu hlasování a úkonů účastníků v jednotlivých fázích.

Fáze SETUP Autorita odešle hlavnímu kontraktu adresy peněženek všech hlasujících voláním `enrollVoters()`. Získání těchto adres a ověření totožnosti hlasujících je v kompetenci autority a není součástí implementace. S ohledem na celkový počet hlasujících může autorita zvolit rozdělení předání adres do několika transakcí, přičemž hlavní kontrakt ověřuje unikátnost obdržených adres. Poté, co hlavnímu kontraktu předá adresy všech oprávněných hlasujících, zahájí autorita rozdělování hlasujících do jednotlivých skupin voláním `splitGroups()`. Rozdělení provádí `MainVotingC` a může opět probíhat postupně ve více transakcích. Požadovaný počet skupin definuje autorita. Kontrakt ke každé z adres hlasujících, které obdržel dříve, přiřadí index skupiny a toto přiřazení uchová ve svém úložišti. V návrhu hlasovacího protokolu je uvažováno rozdělení náhodné, nicméně pro účely implementace demonstrující fungování protokolu jsou hlasující rozděleni deterministicky.

Pro hlasovací skupiny, které vzešly z předchozího rozdělení hlasujících, jsou vytvořeny samostatné kontrakty. Rozdělení vytváření těchto kontraktů do menších transakcí zajišťuje autorita podobně jako v předchozích případech. Vytváření dceřiných kontraktů neprovádí přímo `MainVotingC`, ale využívá k tomu volání `deployBooth()` pomocného kontraktu `VotingBoothDeployer`. V úložišti každého dceřiného kontraktu jsou při jeho vytvoření definovány údaje potřebné pro hlasování – počet hlasujících náležících do dané skupiny, údaje o kandidátech a kryptografické parametry.

Hlavní kontrakt obdrží od pomocného kontraktu adresu vytvořeného dceřiného kontraktu, zaznamená přiřazení adresy dané skupině ve svém úložišti a potvrdí vytvoření kontraktu dané skupiny událostí `BoothCreated()`.

Dceřinému kontraktu nejsou předávány přímo adresy hlasujících. Jednalo by se o nákladnou transakci, přičemž ale dceřiný kontrakt s adresami hlasujících nepracuje. Potřebuje je pouze k ověření, že hlasující, který s ním komunikuje, skutečně náleží do hlasovací skupiny tohoto kontraktu. Je-li potřebné toto ověření, dceřiný kontrakt se dotazuje hlavního kon-



Obrázek 6.2: Průběh hlasování a úkony účastníků v jednotlivých fázích protokolu.

traktu, který ve svém úložišti udržuje mapování validních adres hlasujících na jim přidělené skupiny.

Fáze SIGNUP Jakmile je vytvořen dceřiný kontrakt `VotingBoothC` pro skupinu hlasujících, nachází se tento kontrakt ve fázi `SIGNUP` a je připraven registrovat veřejné klíče hlasujících. Hlasující nejprve získají informace o tom, do které skupiny byli přiřazeni a jaká je adresa příslušného dceřiného kontraktu, z úložiště hlavního kontraktu. Poté již hlasující komunikují pouze s jim přiděleným kontraktem `VotingBoothC`. V této fázi v kontraktu zaregistrují své veřejné klíče. Když jsou v kontraktu zaregistrovány veřejné klíče všech hlasujících dané skupiny, přesune se kontrakt do další fáze. Pokud nejsou všichni hlasující ve fázi `SIGNUP` aktivní a své veřejné klíče neposkytnou, může zahájení další fáze vyvolat kterýkoli z účastníků hlasování.

Fáze PRE_VOTING Účelem fáze `PRE_VOTING` je výpočet veřejných částí hlasovacích klíčů všech hlasujících. Výpočet probíhá v každém z dceřiných kontraktů dle rovnice 6.1 s využitím několika optimalizací, které jsou popsány v sekci 6.4. Výpočet v kontraktech `VotingBoothC` vyvolává autorita a toto volání je delegováno do pomocného kontraktu `VotingFunc`. Hlasující může z kontraktu získat veřejné klíče ostatních hlasujících a veřejnou část hlasovacího klíče si vypočítat lokálně, nebo může z kontraktu získat přímo výsledný klíč po ukončení této fáze, o kterém kontrakty informují událostí `MPCKeysComputedEvent`.

Fáze VOTING Před účastí v této fázi musí hlasující vytvořit svůj hlasovací lístek a vypočítat prvky neinteraktivního zero-knowledge důkazu korektnosti jeho hlasu dle algoritmu 6.3. Svůj hlasovací lístek a odpovídající důkaz odešle kontraktu `VotingBoothC` příslušné skupiny. Kontrakt při převzetí hlasovacího lístku provede ověření připojeného zero-knowledge důkazu (ověření je delegováno opět pomocnému kontraktu `VotingFunc`). Je-li ověření úspěšné, kontrakt hlasovací lístek daného hlasujícího uloží.

Odevzdají-li své hlasovací lístky všichni hlasující, kteří dříve poskytli své veřejné klíče, přesune se kontrakt do fáze `TALLY` a uvědomí o tom i hlavní kontrakt `MainVotingC`. Pokud je však některý z hlasujících v této fázi neaktivní (neodevzdá hlasovací lístek), musí některý z účastníků vyvolat zahájení opravné fáze `FAULT_REPAIR`.

Fáze FAULT_REPAIR Při přechodu do této fáze jsou vyhledáni hlasující, kteří se nezúčastnili odevzdávání hlasovacích lístků. Ostatní hlasující jsou o tomto informováni událostí `MissingVotesEvent`, jejíž součástí jsou indexy neaktivních hlasujících. Každý aktivní hlasující musí vytvořit opravné klíče (jeden za každého neaktivního hlasujícího) pro svůj hlasovací lístek a také důkazy korektnosti těchto klíčů, jak je popsáno v sekci 5.2. Tyto údaje předá příslušnému kontraktu `VotingBoothC` a ten (prostřednictvím pomocného kontraktu `VotingFunc`) provede jak ověření důkazů, tak opravu hlasovacího lístku, který od daného hlasujícího obdržel již ve fázi `VOTING`. O úspěšném provedení opravy každého hlasovacího lístku kontrakt informuje událostí `RepairedBVoteEvent`.

Jakmile jsou opraveny hlasovací lístky všech aktivních hlasujících, přesouvá se kontrakt do fáze `TALLY`, což zároveň oznámí hlavnímu kontraktu.

Fáze TALLY Prvním krokem fáze `TALLY` je výpočet výsledků v jednotlivých dceřiných kontraktech. Tento výpočet, popsán v sekci 5.2, neprovádí přímo jednotlivé kontrakty, ale autorita. Výsledek pak autorita odešle kontraktům k ověření správnosti. Aby mohl kontrakt ověření správnosti výsledku provést, musí sečíst všechny odevzdané hlasovací lístky

v zašifrované podobě. Tento krok je proveden v samostatné transakci (příp. v několika transakcích). Je-li výsledek hlasování v pořádku (odpovídá rovnici 6.2), je odeslán k agregaci do hlavního kontraktu.

Dceřiné kontrakty informují o přechodu do fáze `TALLY` voláním příslušné funkce hlavního kontraktu pomocí `call`. Funkce `call`, podobně jako `delegatecall`, umožňuje interakci mezi kontrakty, v případě `call` se však provedené změny projeví v úložišti cílového kontraktu (v tomto případě `MainVotingC`). Hlavní kontrakt `MainVotingC` přechází do fáze `TALLY`, jakmile obdrží potvrzení přechodu do této fáze od všech dceřiných kontraktů. Poté přijímá výsledky z jednotlivých skupin (opět předávané pomocí `call`) a agreguje je do konečného výsledku. Ten je pro účastníky hlasování dostupný až poté, co jsou do hlavního kontraktu doručeny výsledky všech dceřiných kontraktů. O tom hlavní kontakt informuje událostí `FinalTally`.

6.4 Optimalizace implementace

Ačkoli lepší škálovatelnosti je v návrhu dosaženo především rozdělením hlasování do více smart kontraktů realizujících shodný hlasovací protokol, žádoucí je také dosažení co nejvyššího možného počtu hlasujících v rámci těchto jednotlivých skupin. Za tímto účelem byly v implementaci využity následující optimalizace.

6.4.1 Protokol nad eliptickými křivkami

Návrh protokolu v kapitole 5 využívá problém diskrétního logaritmu definovaného nad cyklickou grupou celých čísel prvočíselného řádu. Pro účely implementace bylo využito problému diskrétního logaritmu nad eliptickými křivkami. Hlavní výhodou eliptických křivek je možnost pracovat s kratšími klíči než v případě celočíselné aritmetiky při zachování stejné úrovně bezpečnosti. Srovnání úrovně bezpečnosti při různých délkách klíčů uvádí tabulka 6.1. V implementaci hlasovacího protokolu nad eliptickými křivkami je využito klíčů délky 256 bitů. Z uvedeného srovnání vyplývá, že pro dosažení stejné úrovně bezpečnosti při implementaci s celočíselnou aritmetikou by bylo nutné použít klíče délky 3072 bitů. Díky použití kratších klíčů je možné snížit jak množství ukládaných a přenášených dat, tak i náročnost výpočtů, což je při využití blockchainové platformy žádoucí, zejména jsou-li cílem co nejnižší náklady a dosažení lepší škálovatelnosti.

Úroveň zabezpečení	Symetrický algoritmus	Asymetrické algoritmy	
		Nad celými čísly	Eliptické křivky
$\leq$ 80 bitů	2TDEA	1024	160–223
112 bitů	3TDEA	2048	224–255
128 bitů	AES-128	**3072**	**256**–383
192 bitů	AES-192	7680	384–511
256 bitů	AES-256	15360	$\geq$ 512

Tabulka 6.1: Srovnání úrovně zabezpečení kryptografických algoritmů při dané délce klíčů (v bitech). Uvažované algoritmy nad celými čísly jsou založeny na faktorizaci čísel (např. RSA) nebo na problému diskrétního logaritmu (např. DSA) [3].

Body eliptické křivky tvoří grupu, která je aditivní (tzn. základní operací je zde sčítání), zatímco protokol s celočíselnou aritmetikou navržený v kapitole 5 je definován nad multipli-

kativní grupou. Tento rozdíl vyžaduje následující úpravu notace navrženého protokolu pro jeho použití s eliptickými křivkami:

1. krok: Parametry hlasování a klíče hlasujících Připadá-li do skupiny N hlasujících a je-li možnost volby mezi k kandidáty, jsou parametry následující:

- generátor G, bod na křivce nad tělesem $\mathbb{F}_p$, kde p je prvočíslo,
- k generátorů $\{F_1, ..., F_k\}$, bodů na křivce generujících stejnou subgrupu jako G.

Každý hlasující P_i vytvoří své klíče, tajný a veřejný. Tajným klíčem je náhodná hodnota $x_i \in_R \mathbb{Z}_n$, kde n je řád bodu G, a veřejným hodnota $G \cdot x_i$.

2. krok: Výpočet hlasovacích klíčů Sdílená část hlasovacího klíče pro každého hlasujícího P_i je vypočítána jako

$$G \cdot y_i = \sum_{j=1}^{i-1} G \cdot x_j - \sum_{j=i+1}^{N} G \cdot x_j \tag{6.1}$$

Samotný hlasovací klíč každého hlasujícího P_i je pak $x_i y_i G$. Platí zde $\sum_i x_i y_i G = O$ [24], kde O je nevlastní bod křivky v nekonečnu.

3. krok: Hlasování Hlasovací lístkem má podobu $V_i = x_i y_i G + F_j$, kde $F_j \in F_1, ..., F_k$ určuje vybraného kandidáta. Spolu s tímto lístkem odešle každý hlasující také neinteraktivní zero-knowledge důkaz dle algoritmu 6.3.

4. krok: Vyloučení neaktivních hlasujících Pokud se někteří z hlasujících nezúčastnili odevzdání hlasovacích lístků, zbývající hlasující vypočítají části svých hlasovacích klíčů, které sdílí s neaktivními hlasujícími. Pro každého aktivního hlasujícího P_i a neaktivního hlasujícího P_j je to hodnota $x_i x_j G$, kde x_i je soukromý klíč aktivního hlasujícího a $x_j G$ je veřejný klíč neaktivního hlasujícího. Zároveň s touto hodnotou poskytne hlasující důkaz její korektnosti dle algoritmu 6.4.

5. krok: Výpočet výsledku Výsledné počty hlasů c_i, $\forall i \in \{1, ..., k\}$ pro jednotlivé kandidáty jsou vypočítány nalezením řešení rovnice 6.2.

$$\sum_{i=1}^{n} V_i = \sum_{i=1}^{n} x_i y_i G + F = c_1 F_1 + c_2 F_2 + ... + c_k F_k \tag{6.2}$$

Pro implementaci protokolu byla zvolena křivka označená jako *secp256k1* definovaná v dokumentu Standards for Efficient Cryptography (SEC) [6].

6.4.2 Souřadnicové systémy

Eliptická křivka může být reprezentována v několika různých souřadnicových systémech. Nejčastěji se setkáváme s afinními souřadnicemi, které reprezentují bod ve tvaru (x, y). Další možností reprezentace jsou homogenní souřadnice v projektivním prostoru, kde je bod definován trojicí (X, Y, Z). Obecný vztah pro převod mezi těmito dvěma reprezentacemi je

Hlasující P_i		*Smart kontrakt*
$H \leftarrow y_i G$		$H \leftarrow y_i G$
Zvolí kandidáta $v_i \in \{1, ..., k\}$		
$F_l \in \{F_1, ..., F_k\}$		
Zveřejní:		
$X \leftarrow x_i G$		
$V_i \leftarrow x_i H + F_l$		
$w \in_R \mathbb{Z}_n$		
$\forall l \in \{1, ..., k\} \setminus v_i$:		
1. $r_l, d_l \in_R \mathbb{Z}_n$		
2. $A_l \leftarrow -d_l X + r_l G$		
3. $B_l \leftarrow r_l H + d_l F_l - d_l V_i$		
for v_i:		
1. $A_{v_i} \leftarrow wG$		
2. $B_{v_i} \leftarrow wH$		
$c \leftarrow H_{zk}(\{\{A_l\}, \{B_l\}\}_l)$		
for v_i:		
1. $d_{v_i} \leftarrow \sum_{l \neq v_i} d_l$		
2. $d_{v_i} \leftarrow c - d_{v_i}$		
3. $r_{v_i} \leftarrow w + x_i d_{v_i}$		
4. $r_{v_i} \leftarrow r_{v_i} \mod n$		
	$\xrightarrow{\forall l: \{A_l\}, \{B_l\}, \{r_l\}, \{d_l\}}$	
		$\Psi \leftarrow \{\forall l : \{A_l\}, \{B_l\}\}$
		$c \leftarrow H_{zk}(\Psi)$
		$\sum_l d_l \stackrel{?}{=} c$
		$\forall l \in \{1, ..., k\}$:
		1. $r_l G \stackrel{?}{=} A_l + d_l X$
		2. $r_l H \stackrel{?}{=} B_l - d_l F_l + d_l V_i$

Obrázek 6.3: Neinteraktivní zero-knowledge důkaz korektnosti hlasovacího lístku při využití eliptických křivek.

Hlasující P_i		*Smart kontrakt*
$A \leftarrow x_i G$, $B \leftarrow x_j G$, x_i		$A \leftarrow x_i G$, $B \leftarrow x_j G$
$w_i \in_R \mathbb{Z}_n$		
$C \leftarrow x_i x_j G$		
$m_1 \leftarrow w_i G$		
$m_2 \leftarrow w_i B$		
$c \leftarrow H_{zk}(A, B, m_1, m_2)$		
$r_i \leftarrow w_i + c x_i$		
	$\xrightarrow{C, r_i, m_1, m_2}$	$c \leftarrow H_{zk}(A, B, m_1, m_2)$
		$rG \stackrel{?}{=} m_1 + cA$
		$rB \stackrel{?}{=} m_2 + cC$

Obrázek 6.4: Neinteraktivní zero-knowledge důkaz korektnosti hodnoty $x_i x_j G$ pro opravný krok hlasování při využití eliptických křivek.

$(x, y) = (X/Z^c, Y/Z^d)$, pro celá čísla c a d. Jeden bod tak může mít několik reprezentací v homogenních souřadnicích (pro různá Z).

Pro eliptické křivky jsou často používanou variantou homogenních souřadnic Jacobiho souřadnice, kde $c = 2$ a $d = 3$. Jacobiho souřadnice jsou využívány především díky efektivnosti operace sčítání bodů. V případě afinních souřadnic je pro sčítání bodů vyžadován náročný výpočet inverzních prvků pro operace v modulární aritmetice. Při použití Jacobiho souřadnic modulární inverze potřebné nejsou, což významně snižuje celkový počet dílčích operací výpočtu.

Knihovna [45] použitá v implementaci umožňuje provádění výpočtů s Jacobiho souřadnicemi a nutnost výpočtu modulárních inverzí je tak možné omezit pouze na situaci, kdy je nutné provést převod z Jacobiho souřadnic zpět na afinní souřadnice (především kvůli možnosti porovnání bodů, které při reprezentaci bodů v Jacobiho souřadnicích není možné přímo provést).

6.4.3 Lokální výpočet modulárních inverzí

Jelikož výpočty protokolu je možné provádět efektivněji reprezentováním bodů v Jacobiho souřadnicích, je převod na afinní souřadnice prováděn až u konečného výsledku. Převod z Jacobiho souřadnic (X, Y, Z) na afinní (x, y) je dán vztahem

$$(x, y) \;=\; (X/Z^2,\, Y/Z^3) \;=\; (X \cdot (Z^{-1})^2,\, Y \cdot (Z^{-1})^3).$$

Jedná se o modulární aritmetiku, inverzní prvek k souřadnici Z lze tedy nalézt pomocí rozšířeného Euklidova algoritmu. Tento výpočet nemusí probíhat na blockchainu, lze jej provést lokálně a na blockchainu pouze ověřit, zda poskytnutý prvek Z^{-1} odpovídá modulární inverzi prvku Z.

Toho je v implementaci využito při výpočtu hlasovacích klíčů, při ověřování korektnosti hlasovacího lístku, při ověřování hodnoty pro opravu hlasovacího lístku a při ověřování výsledku hlasování. Ve všech případech je celý výpočet nejprve proveden lokálně (autoritou u výpočtů hlasovacích klíčů a výsledku, hlasujícími při odevzdání a opravách hlasovacích lístků). Kód pro výpočet je součástí smart kontraktu `VotingCalls`, čímž je zajištěno, aby byl jednotný pro všechny účastníky. Při lokálním výpočtu není následně využit výsledek jako takový (např. sdílená část hlasovacího klíče jednoho hlasujícího), ale je určena hodnota modulární inverze třetí složky (souřadnice Z) výsledného bodu. Tato hodnota je pak předána při volání na blockchainu, kde proběhne opět celý výpočet, ale výsledek může být převeden do afinních souřadnic pomocí předem vypočítaného inverzního prvku, bez nutnosti zahrnout rozšířený Euklidův algoritmus do této transakce.

6.4.4 Násobení bodu skalárem

Naivního přístup k násobení bodu eliptické křivky skalární hodnotou (opakované sčítání) vede na příliš nákladný výpočet pro praktické využití. Při práci s 256 bitovým skalárem by takový výpočet mohl vyžadovat, v nejhorším případě, až $2^{256} - 1$ iterací algoritmu pro sčítání bodů eliptické křivky. Přijatelnějším přístupem je například algoritmus *double-and-add* (příp. jeho varianty), který pracuje s binární reprezentací daného skaláru. V tomto případě však složitost výpočtu závisí na bitové délce skalární hodnoty.

Využitá optimalizace provádí rozložení dané skalární hodnoty na dvě části, každou přibližně poloviční délky [23]. Je-li skalárem číslo $k \in [0...n-1]$ o délce 256 bitů, je číslo k rozloženo na hodnoty k_1 a k_2, z nichž každá je délky 128 bitů, tak že $k = k_1 + k_2\lambda \, mod \, n$,

kde λ je parametr dané křivky. Násobení bodu P skalárem k pak odpovídá rovnici $kP = k_1P + k_2\lambda P \ mod \ n$. K výpočtu lze následně použít algoritmus pro souběžné násobení (*simultaneous multiple point multiplication* [23]). Knihovna Witnet [45] navíc umožňuje výpočet výrazů ve tvaru $kP + lQ$ pomocí souběžného násobení s oběma skaláry k a l rozloženými jako $k = k_1 + k_2\lambda$ a $l = l_1 + l_2\lambda$. V implementaci jsou výpočty odpovídající tvaru $kP + lQ$ součástí ověření výsledků hlasování dle rovnice 6.2. Další výpočty mohou být na tento tvar převedeny. Jedná se o obě rovnice pro ověření důkazu korektnosti hlasu z algoritmu 6.3:

$$r_lG \stackrel{?}{=} A_l + d_lX \quad \Rightarrow \quad A_l \stackrel{?}{=} r_lG - d_lX,$$

$$r_lH \stackrel{?}{=} B_l - d_lF_l + d_lV_i \quad \Rightarrow \quad r_lH + d_lF_l \stackrel{?}{=} B_l + d_lV_i,$$

a stejně také obě rovnice pro důkaz korektnosti opravného klíče z algoritmu 6.4:

$$rG \stackrel{?}{=} m_1 + cA \quad \Rightarrow \quad m_1 \stackrel{?}{=} rG - cA,$$

$$rB \stackrel{?}{=} m_2 + cC \quad \Rightarrow \quad m_2 \stackrel{?}{=} rB - cC.$$

Následující odvození ukazuje, že je-li výraz ve tvaru $kP - lQ$, je možné algoritmus souběžného násobení použít s opačnými čísly k l_1 a l_2:

$$(k_1P + k_2\lambda P) - (l_1Q + l_2\lambda Q) \ = \ (k_1P + k_2\lambda P) + (-l_1Q - l_2\lambda Q)$$

Ve všech těchto případech autorita nebo hlasující nejprve lokálně provede výpočet rozkladu skalárních hodnot pomocí funkce z kontraktu kryptografické knihovny `FastEcMul` a při volání na blockchainu pak předává již rozložený skalár.

6.4.5 Výpočet hlasovacích klíčů

Pokud by výpočet hlasovacích klíčů v kontraktu probíhal zvlášť pro každého hlasujícího (dle rovnice 6.1), docházelo by zde k opakovanému sčítání stejných hodnot. To nevyhnutelně vede k vysokým výpočetním nákladům, které však lze do značné míry eliminovat uchováváním mezivýsledků mezi výpočty jednotlivých klíčů. Výraz pro výpočet hlasovacího klíče lze pro tento účel rozdělit na dvě části.

První částí je menšitel výrazu (pravá strana odčítání), kde dochází ke sčítání veřejných klíčů hlasujících s indexem vyšším než je index hlasujícího, kterému náleží výsledný hlasovací klíč. Hodnoty této části pro jednotlivé hlasující jsou vypočítány ještě před zahájením výpočtu samotných hlasovacích klíčů. Při tomto výpočtu je provedena jedna iterace přes všechny hodnoty veřejných klíčů, přičemž každý veřejný klíč je přičten k průběžnému součtu a uložen do pole menšitelů v úložišti kontraktu.

Druhá (levá) strana výrazu, je pro daného hlasujícího sečtena až při výpočtu hlasovacího klíče. Její hodnota však není zahozena (což by vedlo na opakované sčítání všech veřejných klíčů od nultého indexu), nicméně je uchována a použita pro výpočet hlasovacího klíče následujícího hlasujícího. Pro každý hlasovací klíč je tak na levé straně výrazu provedena jen jedna operace sčítání.

Pouze tyto optimalizace výpočtu hlasovacích klíčů pro dosažení škálovatelnosti nestačí, jelikož probíhají vždy v jediné transakci a při větším počtu hlasujících by došlo k překročení limitu velikosti bloku. Předvýpočet menšitelů i samotný výpočet hlasovacích klíčů je však možné provádět po menších částech, přičemž každá část je provedena v samostatné transakci. Předvýpočet menšitelů je tomuto navíc přizpůsoben tak, že jsou při něm uchovávány pouze počáteční hodnoty pro jednotlivé části výpočtu hlasovacích klíčů, díky čemuž

je využito méně prostoru v úložišti kontraktu. V rámci jedné části je pak hodnota menšitele upravena nanejvýš jednou operací sčítání pro jeden hlasovací klíč, stejně jako v případě levé části výrazu. Mezivýpočet levé strany výrazu je mezi jednotlivými transakcemi uchován v úložišti kontraktu spolu s počátečním indexem pro následující část výpočtu.

Kapitola 7

Vyhodnocení výkonnosti a nákladů

Cílem této kapitoly je zhodnocení vlastností, kterých dosahuje vytvořená implementace hlasovacího protokolu. Hlavní kritéria, která byla při testování implementace sledována jsou především počet hlasujících, které je systém schopen pojmout. S tím souvisí také posouzení, do jaké míry je systém odolný proti selhání hlasování, které může nastat pokud určitý počet zaregistrovaných hlasujících neodevzdá hlasovací lístek. Vyhodnoceny budou také náklady na průběh hlasování a dále bude výsledný systém z této práce srovnán s podobnými, již existujícími přístupy s pohledu výkonnosti i nákladů.

7.1 Testování implementace

Testování implemetace probíhalo s využitím frameworku Truffle Suite[1], který zahrnuje několik nástrojů pro vývoj a testování smart kontraktů pro Ethereum a s ním kompatibilní platformy. Platforma xDai zvolená pro vytvoření implementace v této práci navrženého systému je plně kompatibilní se smart kontrakty pro platformu Ethereum a Truffle Suite tedy bylo možné využít.

Pro účel testů bylo vytvořeno několik modulů v jazyce JavaScript (částečně převzatých z implementace BBB-voting [42]) zajišťujících úkony autority (`authority.js` a `booth.js`) a hlasujících (`voter.js`). Truffle Suite využívá pro testování smart kontraktů knihovnu Mocha. Napsány byly dvě sady testů s využitím této knihovny. První sada, `VotingC.js` simuluje hlasování bez opravné fáze, druhá sada `VotingCfault.js` pak testuje průběh hlasování při neaktivitě některých voličů.

Pro lokální testování byl využit nástroj Ganache z Truffle Suite, který slouží k vytvoření lokálního blockchainu s možností nastavení jeho parametrů. Platforma xDai také nabízí testovací síť POA Sokol, která má stejné parametry jako hlavní síť xDai.

7.2 Dosažené výsledky

Celkový počet hlasujících

Pro samotné hlasování jsou účastníci rozdělováni do skupin v samostatných kontraktech, hlavní kontrakt však musí být schopen pracovat se všemi hlasujícími. Úkolem hlavního kontraktu je před zahájením hlasování účastníky zaregistrovat a rozdělit je do jednotlivých skupin.

[1] https://www.trufflesuite.com

Registrace hlasujících u hlavního kontraktu Nejprve bylo sledováno, kolik adres hlasujících je možné registrovat v jedné transakci. Do jedné transakce (při limitu velikosti bloku 12,5 milionu gas) se podařilo zahrnout až 274 adres hlasujících. Implementace v tomto kroku umožňuje registraci adres po částech, přičemž počet dílčích transakcí omezen není. Tímto krokem tedy není celkový počet hlasujících limitován.

Rozdělení hlasujících do skupin V tomto případě závisí cena transakce na počtu hlasujících, nicméně náročnost výpočtu je také dána zvolenou metodou přiřazování hlasujících do jednotlivých skupin. Při deterministickém rozdělení prováděném v implementaci bylo možné v jedné transakci provést rozdělení ~200 hlasujících. Tato hodnota není limitující z pohledu celkového počtu hlasujících, jelikož i zde implementace umožňuje rozdělení do libovolného počtu transakcí.

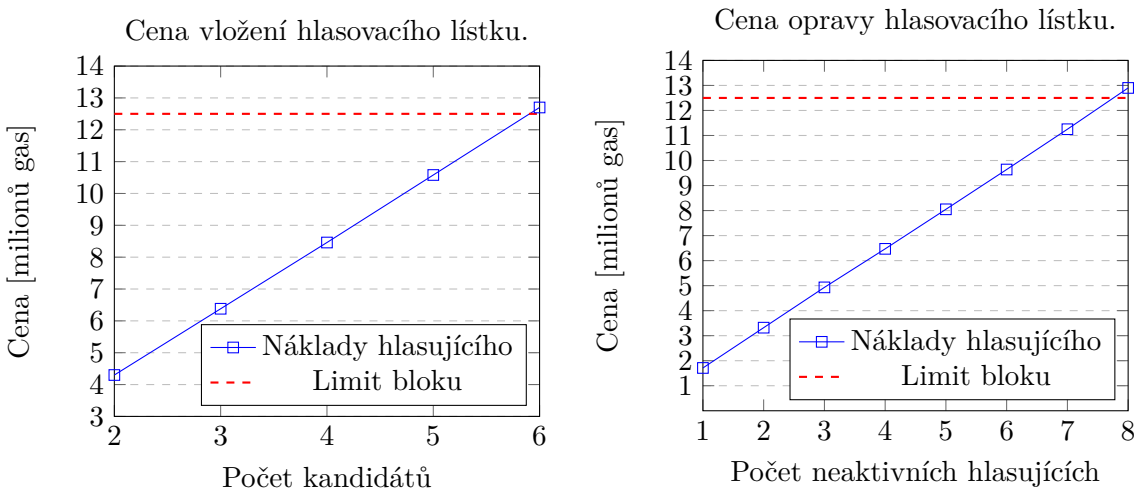
Počet hlasujících v jedné skupině

Jedním z cílů návrhu bylo dosáhnout co nejvyššího možného počtu hlasujících v rámci jedné skupiny (jednoho dceřiného kontraktu). Při testování implementace byla sledována místa, ve kterých by mohla nastat omezení z důvodu závislosti velikosti transakce na počtu hlasujících. Na počtu hlasujících závisí však také náročnost výpočtu výsledku, čímž je i nadále počet hlasujících v jedné skupině limitován.

Výpočet sdílených částí hlasovacích klíčů Také výpočet hlasovacích klíčů může být rozdělen do více transakcí, přičemž v každé z nich lze provést výpočet klíčů až pro 158 hlasujících. Je zde navíc využito optimalizace, kdy část výpočtu je provedena předem (viz sekce 6.4.5), a je-li to nutné, také ve více transakcích. Pokud uvažujeme, že výpočet hlasovacích klíčů bude probíhat pro 150 hlasujících v jedné transakci, pak je možné v jedné transakci předvýpočtu zpracovat až ~1000 veřejných klíčů. Limit objemu transakce předvýpočtu závisí na velikosti částí výpočtu hlasovacích klíčů, jelikož pro každou z těchto částí dochází k provedení nákladné instrukce pro uložení průběžného výsledku předvýpočtu do úložiště kontraktu.

Součet hlasovacích lístků pro ověření výsledku Každý z dceřiných kontraktů ověřuje správnost výsledku hlasování, který byl vypočítán autoritou. Pro toto ověření je nezbytné určení součtu všech odevzdaných hlasovacích lístků. Součet lístků je prováděn v jedné transakci, v níž je možné zpracovat až ~1250 hlasovacích lístků.

Rozdělení sčítání hlasovacích lístků do více transakcí je také součástí testované implementace. Díky tomu je počet hlasujících v jedné skupině limitován pouze náročností výpočtu výsledku (dle rovnice 6.2). Autorita spravující hlasování má možnost zvolit velikost skupin dle svých možností s ohledem na náročnost určení výsledku. Výpočet výsledku je nutné provádět prohledáváním všech možných řešení, což při účasti N hlasujících a k kandidátů představuje $\binom{N+k-1}{k-1}$ možných kombinací [24]. Dle experimentálních výpočtů v [42] lze výsledek s 26,1 mld. možných kombinací vypočítat na běžném laptopu za ~5 hodin. Pro srovnání, v případě 1250 hlasujících (tedy bez využití dávkového zpracování sečtení hlasovacích lístků) a 5 kandidátů je počet možných kombinací ~102,5 mld. Pro zařazení větších počtů hlasujících do jednotlivých skupin by tedy autorita musela disponovat odpovídajícími výpočetními prostředky.



Obrázek 7.1: Nárust ceny odevzdání hlasovacího lístku a opravy hlasovacího lístku v závislosti na počtu kandidátů a neaktivních hlasujících.

Počet skupin

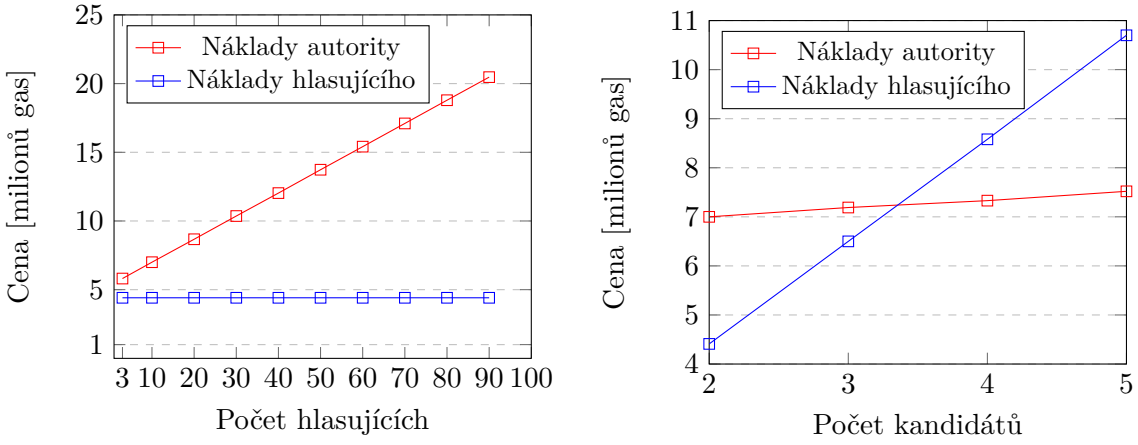
Rozdělení hlasování na menší nezávislé části je zde hlavním konceptem pro dosažení škálovatelnosti hlasovacího protokolu. Je proto žádoucí, aby počet těchto skupin nebyl nijak významně limitován. Problematické je především nákladné vytváření nových dceřiných kontraktů. Z vyhodnocení testů vyplývá, že v jedné transakci je možné vytvořit pouze dva nové kontrakty. Také zde je tedy nutné provedení většího počtu transakcí pro vytvoření potřebných dceřiných kontraktů. Díky tomuto je však možné postupně vytvořit libovolný počet těchto kontraktů, ačkoli se tento způsob řešení škálovatelnosti může projevit vysokými náklady pro autoritu spravující hlasování při vytváření velkého počtu skupin v poměru k počtu hlasujících.

Počet kandidátů

Zero-knowledge důkaz, který kontrakt ověřuje při přijetí hlasovacího lístku od hlasujícího slouží k ověření, zda daný lístek obsahuje korektní volbu jednoho z kandidátů. Jeho složitost tak závisí právě na počtu možností volby (viz algoritmus 6.3). Z testů vyplývá, že nejvyšší možný počet kandidátů je 5, jak ukazuje graf na obrázku 7.1, při vyšším počtu je překročen limit velikosti bloku při odevzdávání hlasu.

Počet neaktivních hlasujících

Při opravě hlasovacích lístků jsou předávány opravné klíče za každého neaktivního hlasujícího a prvky příslušných důkazů. Z grafu na obrázku 7.1 vyplývá, že při předání všech potřebných hodnot v jediné transakci při počtu neaktivních hlasujících vyšším než 7, je velikost transakce opravy hlasovacího lístku nad limitem velikosti bloku. Toto omezení bylo odstraněno postupným opravováním hlasovacího lístku v několika transakcích a systém je tak schopen umožnit opravu při libovolném počtu neaktivních hlasujících, za podmínky, že se opravné fáze zúčastní všichni doposud aktivní hlasující.



Obrázek 7.2: Přehled nárustu nákladů na hlasování v závislosti na počtu hlasujících i kandidátů. Závislost ceny na počtu hlasujících vyhodnocována pro hlasování se dvěma kandidáty, závislost ceny na počtu kandidátu vyhodnocována pro deset hlasujících, v obou případech v jedné hlasovací skupině.

Limitace dané platformou xDai

Ačkoli systém samotný neomezuje počty hlasujících ani možných oprav, je nutné uvažovat objem transakcí, které je možné na blockchainu provést za časovou jednotku. xDai vytváří nový blok každých 5 sekund. Při větších počtech hlasujících (>1000 v každé skupině) většina transakcí hlasovacího systému dosahuje limitní velikosti bloku. Z toho vyplývá, že každých 5 sekund je zpracována jediná transakce hlasovacího systému. Velmi rozsáhlé hlasování by tedy nebylo realizovatelné v přijatelném časovém úseku. Tento problém je ukázán na praktickém příkladu v sekci 7.4.

7.3 Náklady na realizaci hlasování

Většina nákladů na realizaci hlasování je hrazena autoritou, část ale spadá také na jednotlivé hlasující. Hlavní položkou pro autoritu, bez ohledu na počet účastníků hlasování, je vytvoření všech potřebných kontraktů, které vyžaduje ~22 milionů jednotek gas. Dalších ~5 mil. gas autoritu stojí vytvoření každého dceřiného kontraktu po rozdělení hlasujících do skupin. Další náklady jsou pak pro autoritu tvořeny transakcemi, jejichž cena se odvíjí od počtu hlasujících a také od počtu kandidátů. Závislosti ceny na počtu hlasujících i kandidátů ukazují grafy na obrázku 7.2.

Pro každého z hlasujících je celková cena účasti v hlasování dána náklady na registraci veřejného klíče, odevzdání hlasovacího lístku a případně také opravu hlasovacího lístku v případě neúčasti některých z ostatních hlasujících. Zatímco cena registrace veřejného klíče je konstantních 122 tisíc jednotek gas, náklady na odevzdání hlasovacího lístku významně rostou s počtem kandidátů. Za každého kandidáta se cena odevzdání hlasovacího lístku pro každého hlasujícího zvýší o více než 2 mil. jednotek gas. Cena opravy hlasovacího lístku narůstá s počtem neaktivních hlasujících. Přehled dalších nákladů je možné nalézt v tabulce 7.1, kde je uveden příklad hlasování s 1000 hlasujícími a dvěma kandidáty v jedné hlasovací skupině, včetně přibližných přepočtů cen jednotlivých kroků na americké dolary.

	Náklady autority	Náklady hlasujícího
Vytvoření kontraktů		
`EC`	1,36 mil.	
`FastEcMul`	2,62 mil.	
`VotingFunc`	5,66 mil.	
`VotingCalls`	4,2 mil.	
`VotingBoothDeployer`	5,11 mil.	
`MainVotingC`	2,72 mil.	
Registrace hlasujících	45,51 mil.	
Rozdělení do skupin	27,28 mil.	
Vytvoření dceřiného kontraktu	4,85 mil.	
Registrace veřejného klíče		122 tis.
Výpočet hlasovacích klíčů	89,07 mil.	
Odevzdání hlasovacího lístku		4,29 mil.
Přechod do opravné fáze	1,87 mil.	
Oprava hlasovacího lístku		1,67 mil.
Ověření výsledku	9,92 mil.	
Celková cena	178,5 mil.	6,08 mil.
Celková cena v USD	$0,1785	$0,00608

Tabulka 7.1: Příklad nákladů na hlasování v jednotkách gas. Hodnoty odpovídají účasti 1000 hlasujících v jediné skupině (dceřiném kontraktu), jednomu neaktivnímu hlasujícímu a dvěma kandidátům.

7.4 Využití v celostátních volbách

Vzhledem k možnosti zahrnutí velkého počtu hlasujících by bylo možné vytvořený systém využít i pro rozsáhlejší hlasování jako jsou celostátní volby, kde by mohl například usnadnit účast ve volbách občanům pobývajícím v zahraničí nebo zvýšit volební účast mladších voličů. V tomto případě je však limitujícím faktorem značně omezený počet kandidátů (maximálně 5) a také forma hlasování, kde každý hlasující může zvolit jediného kandidáta, bez možnosti zařadit na hlasovací lístek další informace (např. preferenční hlasy). Z těchto důvodů není systém vhodný např. pro parlamentní volby nebo jiná hlasování využívající podobný koncept. Dalším faktorem je také časová náročnost takového hlasování vzhledem k výkonnosti použité platformy xDai. Příkladem vhodné aplikace systému by mohlo být použití pro místní i celostátní referenda, kde jsou obvykle pouze dvě možnosti volby. Jako příklad lze uvést referendum z roku 2003 o vstupu ČR do Evropské unie, které je jediným celostátním referendem, které v ČR proběhlo. Svůj hlas zde odevzdalo 4,5 milionu občanů[2]. Při zavedení možnosti hlasovat elektronicky ve Švýcarsku ze statistik vyplynulo, že přibližně 23 % účastníků se rozhodlo svůj hlas odevzdat on-line [21].

Budeme-li uvažovat tyto hodnoty, mohlo by celostátní referendum s využitím hlasovacího systému z této práce probíhat následovně. Budeme uvažovat, že 1 milion voličů

[2] https://volby.cz/pls/ref2003/re13?xjazyk=CZ

Počet voličů	1 mil.	10 mil.	50 mil.	100 mil.
Limit velikosti bloku (při 1 bloku za 5 s)	125 mil. gas	1,25 mld. gas	6,2 mld. gas	12,5 mld. gas
Limit velikosti bloku (při 1 bloku za 10 s)	248 mil. gas	2,5 mld. gas	12,4 mld. gas	24,8 mld. gas
Rychlost vytváření bloků (max. 12,5 mil. gas na blok)	3 bloky/s	29 bloků/s	145 bloků/s	289 bloků/s

Tabulka 7.2: Vlastnosti, kterých by musela dosahovat blockchainová platforma pro umožnění hlasování s velkými počty voličů (s pouze dvěma kandidáty), kde fáze odevzdávání hlasovacích lístků trvá 2 dny.

(22 % ze 4,5 milionu) se rozhodne hlasovat elektronicky a tito voliči budou rozděleni do 1000 hlasovacích skupin po 1000 hlasujících. Náklady na hlasování by tak dosáhly odhadem 242 miliard jednotek gas pro autoritu a 4,5 milionů jednotek gas pro každého voliče. Celkové cena takového referenda by pak odpovídala částce přibližně 100 tis. korun českých.

Transakce nutné pro realizaci takového hlasování by však vyžadovaly ~520 tis. bloků (při téměř ideálním zaplnění bloků). Vytvoření takového množství bloků na blockchainu xDai by trvalo více než 30 dní. Využití systému v takto rozsáhlém hlasování by tedy bylo z časového hlediska velmi problematické až nepřijatelné.

Aby mohlo takovéto hlasování s jedním milionem voličů proběhnout v obvyklém časovém intervalu, např. 2 dny pouze pro fázi odevzdávání hlasovacích lístků (která je z pohledu množství a objemu transakcí nejnáročnější), musela by platforma umožňovat vytváření až 3 bloků za sekundu, při zachování limitu velikosti bloku 12,5 mil. jednotek gas, nebo umožnit navýšení limitní velikosti bloku. Odhady vlastností blockchainové platformy, které by byly vyžadovány pro realizaci hlasování s velkými počty voličů udává tabulka 7.2.

Jelikož registraci svého veřejného klíče a odevzdávání hlasovacího lístku nemůže hlasující v tomto systému provést zároveň (klíče pro šifrování hlasovacích lístků je možné vypočítat až po registraci všech veřejných klíčů), lze předpokládat, že v reálném použití by nezanedbatelně často docházelo k neaktivitě registrovaných hlasujících. To by vedlo na drahé a opět časově velmi náročné vykonávání opravné fáze, při níž je navíc vyžadována další součinnost voličů (nebo zajištění, aby klientská aplikace na straně voliče byla on-line pro automatizované provedení opravy). Pro řešení tohoto problému je možné vyžadovat povinnou účast hlasujícího, který již registraci provedl, přičemž by mu bylo umožněno odevzdat neutrální hlas, pokud si nebude přát zvolit žádnou z nabízených možností.

7.5 Srovnání s existujícími přístupy

Výsledný hlasovací protokol vychází z přístupů v protokolech Kiayias a Yung [26], Groth [22] a Hao et al. [24]. Stejně jako tyto protokoly tak zajišťuje ochranu soukromí hlasujících, dokonalé utajení hlasování a verifikovatelnost průběhu hlasování. Splňuje také vlastnost self-tallying, nicméně pouze v rámci jednotlivých skupin hlasujících, nikoli pro celkové hlasování. Navíc oproti jmenovaným protokolům umožňuje opravu při selhání (způsobeném neúčastí některých hlasujících) a je škálovatelný. Srovnání s protokoly popsanými v kapitole 3 zobrazuje tabulka 7.3.

	OchS	UtHl	S-T	SlhO	Vrf	Škl
Sensus [11]	✓	×	×	✓	×	✓
Benaloh et al. [5]	✓	×	×	×	✓	×
Cramer et al. [9, 10]	✓	×	×	✓	✓	×
Baudron et al. [4]	✓	×	×	✓	✓	✓
Kiayias a Yung [26]	✓	✓	✓	×	✓	×
Hao et al. [24]	✓	✓	✓	×	✓	×
Groth [22]	✓	✓	✓	×	✓	×
Protokol z této práce	✓	✓	✓*	✓	✓	✓

*pouze v rámci jednotlivých skupin

Tabulka 7.3: Srovnání vlastností existujících hlasovacích protokolů s protokolem vytvořeným v této práci. OchS = ochrana soukromí, UtHl = dokonalé utajení hlasování, S-T = self-tallying, SlhO = odolnost proti selhání, Vrf = verifikovatelnost, Škl = škálovatelnost.

Již existující řešení využívající blockchain jako jsou Open Vote Network [28] a BBB-voting [42] jsou limitovány počtem hlasujících. Pro Open Vote Network je autory doporučeno maximálně 50 hlasujících, hlasování v systému BBB-voting se může účastnit 135 hlasujících. Oba tyto systémy využívají platformu Ethereum a konkrétní hodnoty maximálního počtu účastníků jsou závislé na parametrech (velikosti bloků) Etherea v době, kdy systémy vznikaly, při aktuální velikosti bloku by se řádově jednalo o desítky až stovky hlasujících. Výsledný systém z této práce toto omezení odstranil rozdělením hlasování do menších skupin, ale také dávkovým zpracováním objemných transakcí, zůstává však omezení dané platformou (viz sekce 7.2 a 7.4).

Také díky rozdělení transakce pro opravu hlasovacích lístků v tomto systému není limitován počet neaktivních hlasujících v jednotlivých skupinách (a tedy ani pro celkové hlasování). BBB-voting dokázal hlasovací lístky opravit, pouze pokud se hlasování nezúčastnilo nanejvýš 9 hlasujících, zatímco Open Vote Network opravnou fázi neimplementuje vůbec.

Prezentované řešení zaostává za BBB-voting v možném počtu kandidátů. Zatímco v hlasování s BBB-voting je možné zahrnout až 7 možností volby, prezentovaný systém omezuje hlasování na maximálně 5 kandidátů. Open Vote Network uvažuje pouze hlasování se dvěma variantami volby.

Z pohledu nákladů v jednotkách gas je průběh hlasování v systému z této práce podstatně dražší (přehled uvádí tabulka 7.4), což je dáno větším počtem obsáhlejších kontraktů a transakcí pro dosažení škálovatelnosti. Vzhledem ke značnému rozdílu nákladů na provozování smart kontraktů mezi Ethereem a xDai je však výsledná cena přepočtená na fiat měnu řádově nižší než v případě obou srovnávaných hlasovacích systémů.

	OVN	BBB	tato práce
Vytvoření kontraktů			
Kontrakty hlasování	3,78 mil.	4,8 mil.	17,69 mil.
Kryptografické knihovny	2,44 mil.	2,15 mil.	3,98 mil.
Kontrakt skupiny	–	–	5,74 mil.
Registrace adres hlasujících	2,38 mil.	1,93 mil.	1,86 mil.
Registrace veřejného klíče	0,76 mil.	0,15 mil.	0,12 mil.
Výpočet hlasovacích klíčů	3,09 mil.	2,8 mil.	3,72 mil.
Odevzdání hlasovacího lístku	2,49 mil.	2,72 mil.	4,29 mil.
Ověření výsledku	0,75 mil.	0,39 mil.	0,72 mil.
Celková cena pro autoritu	12,44 mil.	12,07 mil.	33,53 mil.
Celková cena pro autoritu v USD	$2910	$2841	$0,034
Celková cena pro hlasujícího	3,25 mil.	2,87 mil.	4,41 mil.
Celková cena pro hlasujícího v USD	$764	$676	$0,004

Tabulka 7.4: Srovnání nákladů (v jednotkách gas) na hlasování systémů Open Vote Network [28] a BBB-voting [42] na paltformě Ethereum se systémem z této práce na platformě xDai. Hodnoty odpovídají účasti 40 hlasujících a dvou kandidátů. Přepočet na cen na USD v případě Etherea vychází z hodnot z počátku května 2021, tedy $3675 za 1 Ether a průměrné ceny jednotky gas 64 gwei. Hodnota jednoho tokenu xDai odpovídá 1 americkému dolaru, cena jednotky gas je 10^{-9} xDai.

Kapitola 8

Závěr

Cílem práce bylo vytvořit návrh elektronického hlasovacího systému, který splňuje určité požadavky. Požadavky kladené na tento systém byly především ochrana soukromí hlasujících, tedy zajištění toho, aby byl hlas vždy utajen, odolnost systému proti selhání v průběhu hlasování v důsledku neaktivity některých z účastníků a možnost výběru z více než pouze dvou kandidátů. Vyžadována byla také škálovatelnost, která je důležitá pro praktickou využitelnost takového systému, kdy bývá nutné realizovat hlasování s velkým počtem účastníků (např. v celostátních volbách).

Pro nalezení možností realizace takového systému byly zkoumány existující protokoly pro elektronické hlasování. Tyto protokoly byly porovnány na základě jimi dosažených vlastností. Na základě zkoumaných protokolů bylo navrženo řešení hlasovacího systému založeného na smart kontraktech. Protokol je škálovatelný díky rozdělení hlasujících do menších skupin, přičemž v každé z nich je hlasování realizováno samostatným smart kontraktem, a následné agregaci dílčích výsledků nadřazeným smart kontraktem. V rámci jednotlivých skupin protokol dosahuje také dokonalého utajení hlasování a vlastnosti self-tallying.

Navržený protokol je určen pro implementaci využívající smart kontrakty na blockchainu. Nejrozšířenější platformou pro smart kontrakty je Ethereum a právě tuto platformu využívají podobné hlasovací systémy OpenVote Network [28] a BBB-voting [42]. Provozování elektronického hlasování na této platformě je však značně nákladné. U protokolů OpenVote Network [28] a BBB-voting [42] se náklady hlasování na každého účastníka pohybují okolo 700 amerických dolarů, což pro praktické použití není akceptovatelné. Proto byly v rámci této práce zkoumány existující blockchainové platformy pro smart kontrakty a jejich vhodnost pro implementaci vytvořeného návrhu z hlediska nákladů.

Platforma xDai, která byla pro implementaci zvolena, je kompatibilní s platformou Ethereum. Náklady na provozování smart kontraktů na xDai jsou ale v porovnání s Ethereem výrazně nižší, jelikož cena nativního tokenu xDai je stabilní vůči americkému dolaru (1 xDai = $1) a také cena jednotky gas je nízká a konstantní.

Výsledná implementace umožňuje účast mnohonásobně vyššího počtu hlasujících, než u podobných systémů jako např. Open Vote Network [28]. Toho dosahuje dávkovým zpracováním objemných transakcí, rozdělením hlasujících do menších hlasovacích skupin, z nichž pro každou je vytvářen samostatný kontrakt, a optimalizacemi výpočtů hlasovacího protokolu. Nevýhodou tohoto přístupu jsou náklady na hlasování v jednotkách gas. Ty jsou téměř trojnásobně vyšší pro autoritu a o polovinu vyšší pro každého hlasujícího, než u podobných systémů, což se ale díky nízkým cenám jednotky gas na xDai neprojeví ve výsledné ceně hlasování. Problémem to však je z pohledu škálovatelnosti systému. Nový blok je na blockchainu xDai vytvářen každých 5 sekund a je limitován velikostí 12,5 milionu jednotek

gas. S velmi objemnými transakcemi, které jsou nutné pro průběh hlasování při velkých počtech hlasujících, je taková rychlost zpracovávání transakcí značně omezující.

Dalším pokračováním této práce by tak mělo být především snížení objemu transakcí. Nejnákladnější transakce během hlasování jsou vyžadovány především pro ověřování zero-knowledge důkazů při odevzdávání nebo opravách hlasovacích lístků. Zde se jako možné řešení nabízí využití důkazů jako jsou zk-SNARKs nebo zk-STARKs, které nevyžadují provádění náročných výpočtů na blockchainu. Výsledné snížení nákladů by umožnilo nejen účast více hlasujících, ale také více kandidátů. Zavedení této optimalizace a případně také dalších vylepšení jako např. možnosti změny vloženého hlasovacího lístku, by tento systém pro elektronické hlasování mohl být využitelný i v celostátních volbách.

Literatura

[1] Adida, B. Helios: Web-based Open-Audit Voting. In: Oorschot, P. C. van, ed. *Proceedings of the 17th USENIX Security Symposium*. USENIX Association, 2008, sv. 17, s. 335–348. ISBN 978-1-931971-60-7.

[2] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K. et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In: Association for Computing Machinery. *Proceedings of the 13th EuroSys conference*. Duben 2018, s. 1–15. ISBN 9781450355841.

[3] Barker, E. *Recommendation for Key Management*. NIST SP 800-57, Part 1, Rev. 5. Gaithersburg, MD: National Institute of Standards and Technology, 2020.

[4] Baudron, O. et al. Practical multi-candidate election system. In: ACM. *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*. 2001, s. 274–283. DOI: 10.1145/383962.384044. ISBN 1581133839.

[5] Benaloh, J. C. a Yung, M. Distributing the Power of a Government to Enhance the Privacy of Voters. In: Association for Computing Machinery. *Proceedings of the Fifth Annual ACM Symposium on Principles of Distributed Computing*. 1986, s. 52–62. DOI: 10.1145/10590.10595. ISBN 0897911989.

[6] Brown, D. R. L. *Sec 2: Recommended elliptic curve domain parameters*. Certicom Corp., 2010. Dostupné z: https://www.secg.org/sec2-v2.pdf.

[7] *Cardano*. 2020. Dostupné z: https://cardano.org/.

[8] Chaum, D. Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R. L. a Sherman, A. T., ed. *Advances in Cryptology*. Springer US, 1983, s. 199–203. ISBN 978-1-4757-0602-4.

[9] Cramer, R., Franklin, M., Schoenmakers, B. a Yung, M. Multi-Authority Secret-Ballot Elections with Linear Work. In: Maurer, U., ed. *Advances in Cryptology — EUROCRYPT '96*. Springer Berlin Heidelberg, 1996, s. 72–83. ISBN 978-3-540-68339-1.

[10] Cramer, R., Gennaro, R. a Schoenmakers, B. A secure and optimally efficient multi-authority election scheme. In: Fumy, W., ed. *Advances in Cryptology — EUROCRYPT '97*. Springer Berlin Heidelberg, 1997, s. 103–118. DOI: 10.1002/ett.4460080506. ISBN 978-3-540-69053-5.

[11] Cranor, L. F. a Cytron, R. K. Sensus: a security-conscious electronic polling system for the Internet. In: IEEE Computer Society. *Proceedings of the Thirtieth*

Hawaii International Conference on System Sciences. 1997, sv. 3, s. 561–570. DOI: 10.1109/HICSS.1997.661700. ISBN 0818677430.

[12] Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V. et al. BlockChain Technology: Beyond Bitcoin. *Applied Innovation Review*. 1. vyd. Červen 2016, č. 2.

[13] Driza Maurer, A. Updated European Standards for E-voting. In: Krimmer, R. et al., ed. *Electronic Voting*. Springer International Publishing, Leden 2017, s. 146–162. ISBN 978-3-319-68687-5.

[14] *EBallot*. 2020. Dostupné z: https://www.eballot.com.

[15] *Electronic Voting Offers Opportunities and Presents Challenges* [on-line]. United States General Accounting Office, květen 2004 [cit. 2020-30-11]. Dostupné z: https://www.gao.gov/products/GAO-04-766T.

[16] *EOS*. 2021. Dostupné z: https://eos.io.

[17] *Ethereum*. 2020. Dostupné z: https://ethereum.org/.

[18] *The Ethereum Blockchain Explorer*. 2020. Dostupné z: https://etherscan.io/.

[19] *Follow My Vote* [on-line]. 2020. Dostupné z: https://followmyvote.com.

[20] Fujioka, A., Okamoto, T. a Ohta, K. A practical secret voting scheme for large scale elections. In: Seberry, J. a Zheng, Y., ed. *Advances in Cryptology — AUSCRYPT '92*. Springer Berlin Heidelberg, 1993, s. 244–251. ISBN 978-3-540-47976-5.

[21] Gerlach, J. a Gasser, U. Three case studies from Switzerland: E-voting. *Berkman Center Research Publication No*. 1. vyd. 2009, sv. 3, 2009-03.1.

[22] Groth, J. Efficient Maximal Privacy in Boardroom Voting and Anonymous Broadcast. In: Juels, A., ed. *Financial Cryptography*. Springer Berlin Heidelberg, 2004, s. 90–104. ISBN 978-3-540-27809-2.

[23] Hankerson, D., Menezes, A. J. a Vanstone, S. *Guide to elliptic curve cryptography*. 1. vyd. 2004. ISBN 978-0-387-21846-5.

[24] Hao, F., Ryan, P. Y. A. a Zieliński, P. Anonymous voting by two-round public discussion. *IET Information Security*. 1. vyd. 2010, sv. 4, č. 2, s. 62–67. DOI: 10.1049/iet-ifs.2008.0127. ISSN 1751-8717.

[25] *Helios*. 2020. Dostupné z: https://heliosvoting.org.

[26] Kiayias, A. a Yung, M. Self-tallying Elections and Perfect Ballot Secrecy. In: Naccache, D. a Paillier, P., ed. *Public Key Cryptography*. Springer, 2002, s. 141–158. DOI: 10.1007/3-540-45664-3_10. ISBN 978-3-540-45664-3.

[27] Lueks, W., Querejeta Azurmendi, I. a Troncoso, C. VoteAgain: A scalable coercion-resistant voting system. In: Capkun, S. a Roesner, F., ed. *Proceedings of the 29th USENIX Security Symposium*. USENIX Association, 2020, s. 1553–1570. ISBN 978-1-939133-17-5.

[28] McCorry, P., Shahandashti, S. F. a Hao, F. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. In: Kiayias, A., ed. *Financial Cryptography and Data Security*. Springer International Publishing, 2017, s. 357–375. ISBN 978-3-319-70972-7.

[29] Nakamoto, S. *Bitcoin: A peer-to-peer electronic cash system*. 2008.

[30] *Neo*. 2020. Dostupné z: https://neo.org/.

[31] *NEO Scan*. 2020. Dostupné z: https://neoscan.io/.

[32] Nofer, M., Gomber, P., Hinz, O. a Schiereck, D. Blockchain. In: Wiener, M., ed. *Business & Information Systems Engineering*. Springer, 2017, sv. 59, č. 3, s. 183–187. DOI: 10.1007/s12599-017-0467-3. ISSN 2363-7005.

[33] Okamoto, T. Receipt-free electronic voting schemes for large scale elections. In: Christianson et al., ed. *Security Protocols*. Springer Berlin Heidelberg, 1998, s. 25–35. ISBN 978-3-540-69688-9.

[34] Peaster, W. M. *The Current State of Ethereum L2* [on-line]. 2021. Dostupné z: https://defiprime.com/ethereum-l2.

[35] Pressgrove, J. *West Virginia Pauses Use of Voatz Voting App, Cites Security* [on-line]. 2020. Dostupné z: https://www.govtech.com/products/West-Virginia-Pauses-Use-of-Voatz-Voting-App-Cites-Security.html.

[36] *Qtum*. 2021. Dostupné z: https://qtum.org/.

[37] Radwin, M. *An untraceable, universally verifiable voting scheme* [on-line]. 1997. Dostupné z: http://www.radwin.org/michael/projects/voting.pdf.

[38] Schoenmakers, B. A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting. In: Wiener, M., ed. *Advances in Cryptology — CRYPTO' 99*. Springer Berlin Heidelberg, 1999, s. 148–164. ISBN 978-3-540-48405-9.

[39] Seifelnasr, M., Galal, H. a Youssef, A. Scalable Open-Vote Network on Ethereum. In: Bernhard, M., Bracciali, A., Camp, L. J. et al., ed. *Financial Cryptography and Data Security*. Srpen 2020, s. 436–450. DOI: 10.1007/978-3-030-54455-3_31. ISBN 978-3-030-54454-6.

[40] Specter, M. A., Koppel, J. a Weitzner, D. The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in us federal elections. In: Capkun, S. a Roesner, F., ed. *Proceedings of the 29th USENIX Security Symposium*. USENIX Association, 2020, s. 1535–1553. ISBN 978-1-939133-17-5.

[41] Szabo, N. Formalizing and Securing Relationships on Public Networks. *First Monday*. 1. vyd. Září 1997, sv. 2, č. 9. DOI: 10.5210/fm.v2i9.548. Dostupné z: https://firstmonday.org/ojs/index.php/fm/article/view/548.

[42] Venugopalan, S., Homoliak, I., Li, Z. a Szalachowski, P. *BBB-Voting: 1-out-of-k Blockchain-Based Boardroom Voting* [on-line]. 2020. Dostupné z: https://arxiv.org/pdf/2010.09112.pdf.

[43] *Voatz*. 2020. Dostupné z: https://voatz.com.

[44] *VoxVote*. 2020. Dostupné z: https://www.voxvote.com.

[45] *The Witnet Project*. 2021. Dostupné z: https://github.com/witnet.

[46] Wolf, P., Nackerdien, R. a Tuccinardi, D. *Introducing Electronic Voting: Essential Considerations*. 1. vyd. IDEA, 2011. ISBN 978-91-86565-21-3.

[47] *XDai STAKE*. 2021. Dostupné z: https://www.xdaichain.com.

[48] Yaga, D., Mell, P., Roby, N. a Scarfone, K. *Blockchain technology overview*. Oct 2018. Dostupné z: http://dx.doi.org/10.6028/NIST.IR.8202.

[49] Zheng, Z., Xie, S., Dai, H.-N., Chen, X. a Wang, H. Blockchain challenges and opportunities: a survey. *International Journal of Web and Grid Services*. 1. vyd. 2018, sv. 14, č. 4, s. 352–375. DOI: 10.1504/IJWGS.2018.095647.

Příloha A

Obsah přiloženého média

Přiložená SD karta obsahuje následující položky:

- latex/ – zdrojové soubory textu diplomové práce
- vote/ – zdrojové kódy vytvořeného hlasovacího systému
  - build/ – smart kontrakty hlasovacího systému ve zkompilované podobě
  - contracts/ – smart kontrakty hlasovacího systému
  - lib/ – moduly pro testy ve frameworku Truffle
  - migrations/ – skripty pro vytvoření smart kontraktů na blockchainu
  - README.txt – návod pro kompilaci a spuštění
  - test/ – testovací skripty a ukázky výstupů
  - truffle-config.js – konfigurační soubor pro framework Truffle
- xstanc03.pdf – text diplomové práce