Protokoły do obliczeń wielopodmiotowych bazujące na kryptowalutach

streszczenie rozprawy doktorskiej

Marcin Andrychowicz

27 maja 2015

1 Wstęp

Niniejsza rozprawa doktorska składa się z 6 artykułów napisanych podczas studiów doktoranckich na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego:

1. **Secure Multiparty Computations on Bitcoin** [1], przyjęta na *IEEE Symposium on Security and Privacy*, 2014,
2. **Fair Two-Party Computations via the Bitcoin Deposits** [2], przyjęta na *First Workshop on Bitcoin Research (in Assocation with Financial Crypto)*, 2014,
3. **Modeling Bitcoin Contracts by Timed Automata** [3], przyjęta na *Formal Modeling and Analysis of Timed Systems (FORMATS)*, 2014,
4. **On the Malleability of Bitcoin Transactions** [4], przyjęta na *Second Workshop on Bitcoin Research (in Assocation with Financial Crypto)*, 2015,
5. **Secure Multiparty Computations on Bitcoin** [5], przyjęta do *Communications of the ACM (CACM), Research Highlights*, 2015,
6. **PoW-Based Distributed Cryptography with no Trusted Setup** [6], przyjęta na *International Cryptology Conference (CRYPTO)*, 2015.

Pierwsze 5 artykułów napisałem wspólnie z promotorem drem hab. Stefanem Dziembowskim oraz mgr Danielem Malinowskim i mgr Łukaszem Mazurkiem. Współautorem ostatniej pracy jest jedynie dr hab. Dziembowski. Wszystkie artykuły dotyczą zastosowań walut kryptograficznych (głównie Bitcoina) oraz technik wywodzących się z kryptowalut do konstrukcji protokołów do obliczeń wielopodmiotowych (ang. Multiparty Computation Protocols, MPC), które wychodzą poza standardową definicję bezpieczeństwa protokołów MPC.

Krótkie wprowadzenie do protokołów MPC zamieszczono w rozdziale 2, zaś wstęp do tematyki kryptowalut i Bitcoina znajduje się w rozdziale 3. Rozdział 4 przedstawia wyniki opublikowane w artykułach wchodzących w skład niniejszej rozprawy. W rozdziale 5 zamieściłem listę moich artykułów opublikowanych podczas studiów doktoranckich, które nie weszły w skład niniejszej rozprawy.

2 Protokoły do bezpiecznych obliczeń wielopodmiotowych

Protokoły do bezpiecznych obliczeń wielopodmiotowych (ang. MPC, [7, 8]) pozwalają wzajemnie nieufającym sobie stronom na obliczenie wartości dowolnej funkcji f na ich prywatnych i tajnych wejściach bez ujawniania ich wartości. Dokładniej wymagamy, aby żadna ze stron nie dowiedziała się więcej o wejściach pozostałych stron, niż mogłaby się dowiedzieć w *idealnym modelu*, w którym: (a) wszystkie strony przesyłają swoje wejścia do zaufanej trzeciej strony (ang. trusted third party); (b) zaufana trzecia strona oblicza wartość funkcji f; i (c) zaufana trzecia strona przesyła obliczoną wartość funkcji f wszystkim stronom protokołu.

Jako ilustrację definicji bezpieczeństwa protokołów MPC rozważmy tzw. *problem małżeństwa*. W problemie tym występują dwie strony, które będziemy nazywać Alicja i Bartosz. Nasi bohaterowie chcą sprawdzić, czy oboje chcieliby się pobrać. Wejściem Alicji jest bit a, który oznacza, czy chce ona wyjść za Bartosza, zaś wejściem Bartosza jest bit b oznaczający, czy chce on ożenić się z Alicją. Wynikiem protokołu jest bit oznaczający, czy *obie* strony chcą się pobrać, czyli funkcja której wartość liczymy to koniunkcja: $f(a, b) = a \wedge b$. Zauważmy, że jeśli $a = 1$, to Alicja może łatwo wywnioskować wartość bitu b z wyniku protokołu, gdyż $f(a, b) = a \wedge b = b$. Tajność wejść oznacza w tym przypadku, że jeśli Alicja nie chce wyjść za Bartosza (tzn. $a = 0$), to nie powinna ona poznań wartości bitu b i analogiczny warunek powinien zachodzić dla Bartosza.

Pomimo olbrzymich możliwość teoretycznych protokoły MPC są rzadko używane w praktyce. Jedną z przyczyn takiego stanu rzeczy jest fakt, że protokoły takie nie gwarantują w ogólności tzw. *sprawiedliwości obliczeń* (ang. fairness), co oznacza, ze nieuczciwa strona może przerwać wykonanie protokołu w stanie, w którym zna ona wynik protokołu, a pozostałe strony nie znają go. Wiadomo, że sprawiedliwości obliczeń nie można zagwarantować bez założenia, że większość stron wykonujących protokół jest uczciwa. W szczególności protokoły MPC dla dwóch stron nie gwarantują sprawiedliwości obliczeń (o ile nie założymy, że obie strony są uczciwe).

3 Waluty kryptograficzne

Waluty kryptograficzne (nazywane również kryptowalutami) są środkami płatniczymi, które mogą być używane za pośrednictwem internetu. *Bitcoin* [9] jest kryptowalutą stworzoną w 2008 roku roku przez anonimowego developera ukrywającego się pod pseudonimem „Satoshi Nakamoto". Bitcoin jest pierwszą kryptowalutą, która zyskała dużą popularność — obecna kapitalizacja rynkowa Bitcoina przekracza 3 mld USD (maj 2015). Najważniejszą cechą Bitcoina odróżniającą go od tradycyjnych walut jest fakt, że nie istnieje żadna instytucja, która kontroluje infrastrukturę Bitcoina lub która mogłaby „wydrukować" więcej bitcoinów. Ponadto opłaty transakcyjne w Bitcoinie są bardzo niskie i zapewnia on pewien poziom anonimowości.

Błyskawiczny sukces Bitcoina spowodował powstanie wielu innych kryptowalut (nazywanych *kryptowalutami alternatywnymi* — ang. *altcoins* — dla odróżnienia od Bitcoina). Większość alternatywnych kryptowalut nie różni się w znaczący sposób od Bitcoina. W niniejszej rozprawie koncentruję się na Bitcoinie jako najbardziej popularnej kryptowalucie, ale proponowane rozwiązania są uniwersalne i dotyczą idei stojących za większością kryptowalut.

Temat Bitcoina jest niezwykle interesujący z punktu widzenia kryptografii, ale z braku miejsca w ninejszym streszczeniu opiszę jedynie cechy Bitcoina, które są najbardziej związane z moimi badaniami. Bardziej szczegółowy opis Bitcoina można znaleźć w moich pracach, np. [5].

Wszyscy użytkownicy Bitcoina tworzą sieć peer-to-peer, która utrzymuje listę wszystkich przelewów (zwaną *blockchainem*), które miały miejsce w systemie. Przypomnijmy, że tradycyjne protokoły MPC nie gwarantuję sprawiedliwości obliczeń, jeśli nie założy się, że większość stron jest uczciwa. Założenie takie jest praktycznie niemożliwe do zapewnienia w internecie, ponieważ przeciwnik może wówczas stworzyć wiele fałszywych tożsamości (ang. *Sybil attack*).

Bitcoin rozwiązuje ten problem poprzez użycie tzw. *Proofs-of-Work (PoW)*,

które są narzędziami kryptograficznymi pozwalającymi jednej stronie udowodnić pozostałym stronom, że wykonała ona pewną liczbę operacji (np. obliczeń wartości zadanej funkcji haszującej) na zadanym problemie. Konkretniej bezpieczeństwo Bitcoina opiera się na założeniu, że większość mocy obliczeniowej w systemie jest kontrolowana przez uczciwych użytkowników. Aby złamać bezpieczeństwo systemu, przeciwnik musiałby zatem posiadać moc obliczeniową zbliżoną do mocy obliczeniowej wszystkich uczciwych użytkowników Bitcoina.

Jak już wspomniano, sieć złożona ze wszystkich użytkowników Bitcoina utrzymuje blockchain zawierający wszystkie transakcje, które miały miejsce w systemie od czasu jego powstania. W pierwszym przybliżeniu można powiedzieć, że transakcja w Bitconie ma postać „Użytkownik A przelewa x bitcoinów użytkownikowi B". Oczywiście konieczny jest jakiś mechanizm weryfikujący, czy użytkownik A posiada odpowiednią liczbę bitcoinów do realizacji takiego przelewu. W tym celu każda transakcja posiada specjalny ciąg identyfikujący ją (w rzeczywistości jest to wartość pewnej funkcji haszującej na jej binarnej reprezentacji) i ma tak naprawdę postać „Użytkownik A przelewa x bitcoinów, które otrzymał wcześniej w transakcji T, użytkownikowi B". Taka transakcja jest akceptowana, jedynie jeśli blockchain zawiera transakcję T i transakcja T przelewa x bitcoinów do użytkownika A. W tym przypadku mówimy, że nowa transakcja *wydaje* transakcję T. Musimy oczywiście upewnić się, że nieuczciwy użytkownik nie może wydać swoich bitcoinów dwukrotnie, więc nowa transakcja jest akceptowana jedynie w przypadku, kiedy blockchain nie zawiera innej transakcji wydającej transakcję T.

Fakt posiadania bitcoinów jest udowadniany za pomocą kryptografii klucza publicznego (każdy użytkownik sam generuje swój klucz prywatny i publiczny). Dokładniej użytkownicy systemu są identyfikowani za pomocą ich kluczy publicznych a każda transakcja zawiera klucz publiczny odbiorcy. Transakcja wydająca transakcję T jest akceptowana tylko pod warunkiem, że jest podpisana używając klucza prywatnego odpowiadającemu kluczowi publicznemu odbiorcy transakcji T.

Ponadto, co okaże się niezwykle istotne z punktu zastosowania Bitcoina do konstrukcji protokołów MPC, możliwe są bardziej zaawansowane typy transakcji (zwane transakcjami *niestandardowymi*). Takie transakcje mogą zawierać rodzaj zagadki, którą odbiorca musi rozwiązań, aby przelew doszedł do skutku. Dokładniej każda transakcja zawiera tzw. *skrypt wyjściowy*, który jest opisem pewnej funkcji f. Aby przelew doszedł do skutku, odbiorca musi podać wartość x dla której $f(x) = \mathbf{true}$. Funkcja f jest pisana w specjalnie stworzonym do tego celu języki programowania, który zawiera m.in. instrukcje do weryfikowania podpisów kryptograficznych i liczenia wartości pewnych funkcji haszujących. Ponadto transakcje mogą zawierać

blokady czasowe (ang. time-locks), co oznacza, że bitcoiny, które są w nich przesłane mogą zostać odebrane dopiero po pewnym czasie. Przykładowo możliwe jest wysłanie bitcoinów do trzech osób w taki sposób, że każde dwie z nich mogą odebrać bitcoiny po 1 lipca 2015 roku pod warunkiem, że podadzą łańcuch znaków x, taki że $H(x) = v$, gdzie H jest funkcją haszującą SHA-256 a v to pewny łańcuch znaków.

4 Uzyskane wyniki

4.1 Bezpieczne kasyno internetowe

Internetowy hazard stał się w ostatnim czasie bardzo popularny — szacuje się, że działa obecnie 1700 kasyn internetowych obsługujących zakłady warte ponad 4 mld USD rocznie [10]. Klienci kasyn internetowych nie są w żaden sposób chronieni przed nadużyciami ze strony kasyn. Kasyno może łatwo oszukać swoich klientów poprzez np. użycie losowości z innego rozkładu niż założony lub po prostu odmawiając użytkownikowi wypłaty należnej mu nagrody. Znanych jest wiele przypadków, kiedy kasyna internetowe oszukiwały swoich klientów.

Naturalne wydaje się więc pytanie, czy nie można zastosować protokołów MPC do zapewnienia klientów kasyn, że gra będzie uczciwa. Dla uproszczenia rozważmy przypadek dwóch stron (jedna z nich może być na przykład kasynem internetowym), które nie ufają sobie wzajemnie i chciałyby postawić 1zł na rzut wirtualną monetą — innymi słowy obie strony wpłacają po 1zł i jedna z nich, wybrana losowo, otrzymuje 2zł. Istnieją protokoły MPC, które pozwalają wylosować bit z rozkładu jednostajnego, ale czy można zastosować je w powyższej sytuacji? Okazuje się, że protokoły te nie dają się zastosować w ten sytuacji z dwóch powodów. Po pierwsze, protokoły MPC dla dwóch stron nie gwarantują sprawiedliwości obliczeń, co oznacza, że nieuczciwa strona mogłaby przerwać wykonanie protokołu w stanie, w którym jedynie ona wie kto wygrał. Dodatkowy problem stanowi fakt, że nawet jeśli obie strony wiedzą, kto wygrał, to nie ma możliwości zapewnić, że gracz, który przegrał, zapłaci uzgodnioną złotówkę zwycięzcy.

W naszej pierwszej pracy [1] (i jej wersji czasopismowej [5]) pokazaliśmy, że oba powyższe problemy można rozwiązać, zakładając, że strony mają dostęp do infrastruktury Bitcoina. Dokładniej zaprezentowaliśmy konstrukcję protokołu dla dowolnej liczby stron, który pozwala stronom obstawiać rzuty wirtualnych monet. Protokół może być użyty w sieci peer-to-peer a jego użytkownicy mogą zachować anonimowość. Nasze rozwiązanie daje bardzo silną gwarancję bezpieczeństwa — bez względu na liczbę nieuczciwych stron i ich

zachowanie uczciwe strony mogą być pewne, że gra będzie przeprowadzona w uczciwy sposób.

Główna idea nowego protokołu jest następująca. Na początku protokołu każda ze stron robi specjalny *depozyt* w bitcoinach. Depozyt jest pewną niestandardową transakcją Bitcoinową zaprojektowaną w taki sposób, że strona będzie mogła odzyskać depozyt, jedynie jeśli będzie postępowała zgodnie z protokołem. Jeśli nieuczciwa strona spróbuje oszukiwać, wówczas jej depozyt zostanie automatycznie oddany pozostałym uczestnikom protokołu, rekompensując im ewentualnie straty powstałe na skutek oszustwa.

W [1] nie tylko zaprezentowaliśmy wspomniany protokół, ale przede wszystkim zaproponowaliśmy nowy paradygmat obliczeń polegający na połączeniu protokołów MPC i kryptowalut.

4.2 Sprawiedliwe protokoły MPC z konsekwencjami finansowymi

W naszej drugiej pracy [2] uogólniliśmy wyniki z [1] i przedstawiliśmy konstrukcję protokołu MPC dla dwóch stron dla dowolnej funkcji f, który posiada dwie zalety w sposunku do klasycznych protokołów MPC.

Po pierwsze, nasz protokół gwarantuje pewną formę sprawiedliwości obliczeń. Dokładniej obie strony przed rozpoczęciem protokołu robią depozyty w bitcoinach. Jeśli obie strony postępują zgodnie z protokołem, to wówczas poznają one wartość funkcji f i otrzymują swoje depozyty z powrotem. Jeśli jednak jedna ze stron nie będzie postępować zgodnie z protokołem (to znaczy przerwie wykonywanie protokołu lub wyśle niepoprawną wiadomość), to wówczas druga strona otrzymuje *oba* depozyty. Innymi słowy, strona, która nie pozna wyniku obliczenia, otrzyma rekompensatę finansową w bitcoinach. Możemy zatem powiedzieć, że protokół ten gwarantuje sprawiedliwość obliczeń, o ile wysokość depozytu przewyższa stratę spowodowaną niepoznaniem wyniku obliczenia. Ponadto wykowanie takich depozytów jest zupełnie bezpieczne — strona, która postępuje zgodnie z protokołem, ma pewność, że otrzyma ona swój depozyt bez względu na postępowanie drugiej strony.

Drugą zaletą naszego protokołu jest fakt, że wartość funkcji f, którą liczymy może zawierać instrukcje w stylu "Alicja płaci Bartoszowi 1 bitcoina" i nasz protokół gwarantuje, że takie przelewy będą automatycznie wykonane. Oczywiście przelew zostanie wykonany tylko, jeśli protokół zostanie przeprowadzony do końca, ale jeśli jedna ze stron przerwie jego wykonanie, to jej depozyt zostanie przekazany drugiej stronie. Protokół ten może zostać m.in. wykorzystany do handlu dobrami elektronicznymi. Ponadto pozwala on na sprzedaż informacji w taki sposób, że nawet sprzedawca nie wie dokładnie,

jaką informację sprzedaje. Szczegóły można znaleźć w naszej pracy.

Nasz protokół z [2] wymagał drobnej modyfikacji protokołu Bitcoina związanej z tzw. *kowalnością* (ang. malleability) transakcji, która jest dokładniej opisana w następnym rozdziale.

4.3 Problem kowalności transakcji

Transakcje w Bitcoinie są kowalne, co oznacza, że mając daną transakcję T, każdy użytkownik systemy może stworzyć transakcję T', która jest semantycznie równoważna T (ma tego samego nadawcę, odbiorcę i kwotę), ale różni się od T reprezentacją binarną. Może to powodować poważne problemy, ponieważ infrastruktura Bitcoina śledzi transakcje używając haszy ich binarnej reprezentacji a transakcje są rozgłaszane w sieci peer-to-peer. W [4] eksperymentalnie sprawdziliśmy na ile poważny jest ten problem. Pokazaliśmy, że przeciwnik może relatywnie łatwo przeprowadzić atak polegający na podmienieniu binarnej reprezentacji transakcji wysłanej przez innego użytkownika. Sprawdziliśmy, jak zachowuje się większość dostepnych klientów Bitcoina (programów, używanych do wysyłania Bitcoinów) w tej sytuacji, i odkryliśmy szereg błędów w oprogramowaniu. Błędy te mogą powodować, że użytkownik nie będzie mógł (przynajmniej tymczasowo) dokonywać żadnych przelewów, co oznacza, że efektywnie traci on wszystkie posiadane bitcoiny.

Ponadto kowalność transakcji stanowi problem dla większości kontraktów wykorzystujących Bitcoina. W [4] zaprezentowaliśmy również ogólną metodę ochrony konktraktów przed atakami związanymi z tym problemem. W szczególności pozwala ona zastosować nasz protokół z [2] w obecnej wersji protokołu Bitcoina.

4.4 Formalna weryfikacja kontraktów

Protokoły wykorzystujące Bitcoina (zwane też *kontraktami*) są trudne do analizy z powodu rozproszonej natury blockchaina i dużej liczby możliwych przeplotów. Ponadto błędy w konstrukcjach takich protokołów mogą zostać wykorzystane przez nieuczciwe strony dla uzyskania korzyści finansowych.

Z tego powodu wartościowe byłoby wykorzystanie metod weryfikacji formalnej do analizy takich protokołów. W [3] zaprezentowaliśmy framework do formalnego modelowania kontraktów wykorzystujących Bitcoina za pomocą automatów z czasem. Zaprezentowana metoda jest ogólna i może być zastosowana do formalnej weryfikacji prawie dowolnych kontraktów. Nasz framework pozwala weryfikować własności w stylu "W rezultacie wykonania protokoły strona zyskuje 1 bitcoina lub poznaje ciąg znaków s spełniający

warunek $C(s)$ bez względu na postępowanie drugiej strony". Ponadto, używając naszego frameworku, zweryfikowaliśmy poprawność i bezpieczeństwo dwóch kontraktów z naszych poprzednich prac [1, 4] w programie UPPAAL [11].

4.5 Rozproszona kryptografia oparta na Proofs-of-Work

Podejście opisane dotychczas polegało na projektowaniu protokołów, w których strony mają dostęp do pewnej kryptowaluty. Możliwe jest również wzbogacenie protokołów MPC za pomocą technik pochodzących z kryptowalut w zupełnie inny sposób. Jednym z innowacyjnych pomysłów, stojących za kryptowalutami, było wykorzystanie założenia, że większość mocy obliczeniowej należy do uczciwych użytkowników, gdzie o mocy obliczeniowej danego użytkownika świadczy to, ile razy może on policzyć wartość pewnej funkcji haszującej (np. SHA-256) w danym przedziale czasu.

W [6] rozważaliśmy protokoły kryptograficzne w sieci peer-to-peer przy założeniu, że moc obliczeniowa przeciwnika jest ograniczona i nie ma zapewnionej żadnego typu infrastruktury w stylu Infrastruktury Klucza Publicznego (ang. Public Key Infrastructure, PKI) lub źródła skorelowanej losowości (ang. random beacon); nawet liczba stron wykonujących protokół nie jest znana stronom. Zaproponowaliśmy formalny model dla tej sytuacji i zaprezentowaliśmy dwa protokoły w tym modelu.

Pierwszy protokół to protokół do rozgłaszania niezawodnego (ang. reliable broadcast), który działa przy założeniu, że znane jest górne ograniczenie π na łączną moc obliczeniową wszystkich stron oraz, że moc obliczeniowa należąca do uczciwych stron jest niezaniedbywalną częścią π.

Drugi protokół działa przy założeniu, że większość mocy obliczeniowej w sieci jest kontrolonawa przez uczciwe strony i pozwala na obliczenie podzbioru stron G, takiego, że (1) wszystkie strony znają zbiór G (konsensus); (2) wszystkie uczciwe strony należą do G; (3) większość stron w G jest uczciwa; i (4) każda strona zna klucz publiczny każdej strony ze zbioru G. Istnienie takiego protokołu implikuje, że każda funkcjonalność (np. porozumienie Bizantyjskie), która może zostać zrealizowana przy klasycznym założeniu, że większość stron jest uczciwa oraz że dostępna jest infrastruktura PKI, może zostać również zrealizowana przy założeniu, że większość mocy obliczeniowej należy do uczciwych stron *bez* dodatkowego założenia o PKI.

5 Inne publikacje

Podczas studiów doktoranckich opublikowałem również poniższe dwa artykuły. Nie stanowią one części niniejszej rozprawy, ponieważ nie dotyczą one protokołów MPC, ani kryptowalut.

1. **Efficient Leakage Resilient Circuit Compilers**, przyjęta na *RSA Conference Cryptographers' Track (CT-RSA)*, M. Andrychowicz, I. Damgaard, S. Dziembowski, S. Faust, A. Polychroniadou, 2015,

2. **Leakage-Resilient Cryptography over Large Finite Fields: Theory and Practice**, przyjęta na *International Conference on Applied Cryptography and Network Security (ACNS)*, M. Andrychowicz, D. Masny, E. Persichetti, 2015.

Literatura

[1] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Secure multiparty computations on bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 443–458. IEEE, 2014.

[2] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Fair two-party computations via bitcoin deposits. In *Financial Cryptography and Data Security*, pages 105–121. Springer, 2014.

[3] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Modeling bitcoin contracts by timed automata. In *Formal Modeling and Analysis of Timed Systems*, pages 7–22. Springer, 2014.

[4] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. On the malleability of bitcoin transactions. In *Workshop on Bitcoin Research*, 2015.

[5] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Secure multiparty computations on bitcoin. In *Communications of the ACM, Reserach Highlights*. ACM, 2015.

[6] Marcin Andrychowicz and Stefan Dziembowski. Pow-based distributed cryptography with no trusted setup. In *International Cryptology Conference*. Springer, 2015.

[7] A. C.-C. Yao. How to generate and exchange secrets. In *FOCS*, 1986.

[8] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. STOC, 1987.

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

[10] The Economist. Online gambling: Know when to fold, 2013.

[11] Gerd Behrmann, Re David, and Kim G. Larsen. A tutorial on uppaal 4.0, 2006.