Kryptografia

Proof-of-concept gry społecznościowej opartej o Anonymous Veto Network

Maciej Kabała, Paweł Narolski

18 czerwca 2020 r.

1 Przedstawienie projektu

Prawo do wolności słowa: nieskrępowanego wyrażania własnych opinii (np. krytyki) na dany temat stanowi jedną z fundamentalnych, demokratycznych wartości.

Niestety, nie każdy obywatel w każdej dowolnej sytuacji może pozwolić sobie na przedstawianie swoich własnych poglądów w sposób otwarty i szczery. Dzieje się tak nie tylko, gdy jest on mieszkańcem państwa autorytarnego. W wielu, codziennych i przyziemnych sytuacjach, jak np. podczas dyskusji z osobą, od której jest on zależny, i z którą się nie zgadza, statystyczny obywatel woli nie wyrażać swoich własnych poglądów [1].

Celem naszego projektu jest zaproponowanie rozwiązania opartego o *Anonymous Veto Network* [2], które umożliwi udzielenie statystycznemu obywatelowi szczerej odpowiedzi "tak" lub "nie" na zadane pytanie - bez obawy o związane z tym reperkusje - jednocześnie uniemożliwiając komukolwiek innemu znalezienie odpowiedzi na inne pytanie: "kto (i czy to ten obywatel, o którym myślę) się ze mną nie zgodził".

Proof-of-concept naszego rozwiązania będzie stanowiła prosta gra społecznościowa, której użytkownicy będą mogli udzielać odpowiedzi na losowo wybrane pytania. Dzięki autorskiej implementacji protokołu *AV-Net* (opisanego w rozdziale 2) żaden uczestnik gry (lub adwersarz przyglądający się rozgrywce graczy) nie będzie mógł ustalić, jak dana osoba odpowiedziała na zadane pytanie, a jednocześnie każdy z graczy będzie mógł się dowiedzieć, czy któryś z nich nie zgodził się z pozostałymi uczestnikami rozgrywki.

Nasza aplikacja zostanie przygotowana w języku `Go` w modelu klient-serwer, w którym wykorzystane zostanie szyfrowanie `mTLS`. Serwer będzie synchronizował stan gry pomiędzy jej uczestnikami - graczami - pełniąc jednocześnie rolę bezpiecznego kanału wymiany informacji pomiędzy uczestnikami; rozwiązanie to zostało podyktowane względami praktycznymi (standardowo, w celu uniknięcia problemów związanych z synchronizacją stanu rozgrywki, wymiana informacji na temat stanu gier sieciowych nie odbywa się w modelu *peer-to-peer*).

Rezultatem końcowym naszego projektu będą gotowe aplikacje klienckie oraz serwerowe, które po skompilowaniu będą mogły zostać uruchomione na najważniejszych platformach systemowych (Linux, macOS, Windows) w formie aplikacji dostarczających prosty, tekstowy interfejs użytkownika.

Nasze rozwiązanie będzie możliwe do rozwinięcia w pełnoprawny produkt: grę on-line realizowaną w modelu klient-serwer po stworzeniu graficznego interfejsu użytkownika, lub inne rozwiązanie komercyjne umożliwiające udzielanie anonimowych odpowiedzi na zadane pytania zamknięte.

2 Anonymous Veto Network

W naszym projekcie wykorzystany został protokół Anonymous Veto. Składają się na niego dwie rundy, które zostały poniżej opisane. Niech udział w głosowaniu bierze n uczestników, oznaczonych jako P_i, $i \in \{1, ..., n\}$. Na starcie niech dana będzie skończona cykliczna grupa G o rzędzie q, gdzie q jest liczbą pierwszą. W grupie

tej problem Decisional Diffie-Hellman (DDH) musi być trudny. Niech dany będzie generator g grupy G. Każdy z graczy ma dostęp do następujących danych: G, g, q. Uczestnikom przedstawione zostaje pytanie i rozpoczyna się głosowanie.

2.1 Runda pierwsza

Każdy z użytkowników P_i wybiera pseudolosową liczbę (mającą pozostać tajną) x_i taką, że

$$x_i \in Z_q.$$

Następnie uczestnicy wysyłają do serwera liczbę g^{x_i}, która zostanie przekazana wszystkim pozostałym osobom. Po wykonaniu tego kroku każdy z użytkowników posiada wszystkie liczby g^{x_i}. Ostatni etap, to policzenie g^{y_i}, gdzie

$$g^{y_i} = \frac{\prod_{k=1}^{i-1} g^{x_k}}{\prod_{k=i+1}^{n} g^{x_k}}.$$

Każdy z uczestników wyliczoną wartość zachowuje dla siebie.

2.2 Runda druga

W tej rundzie użytkownik decyduje, czy zgadza się z decyzją (no veto) czy też chce ją zawetować (veto). Rozważmy zatem oba scenariusze.

2.2.1 Brak weta

W tym przypadku uczestnik oblicza wartość v_i, gdzie

$$v_i = g^{x_i y_i}.$$

Tak obliczona wartość zostaje przesłana do serwera i rozesłana innym uczestnikom.

2.2.2 Weto

Jeśli gracz chce zawetować decyzję, to oblicza wartość v_i inaczej. Najpierw wybiera pseudolosową liczbę r_i taką, że

$$r_i \in Z_q, r_i \neq x_i.$$

Wtedy v_i przyjmuje wartość

$$v_i = g^{r_i y_i}.$$

Tak obliczona wartość zostaje przesłana do serwera i rozesłana innym uczestnikom.

2.3 Rozstrzygnięcie

Po otrzymaniu wszystkich wartości v_i serwer rozsyła je do graczy. Następnie każdy z nich może samemu sprawdzić, czy ktokolwiek zawetował w głosowaniu. W tym celu obliczany jest produkt $\prod_i v_i$. Jeżeli

$$\prod_i v_i = 1,$$

to nikt nie zgłosił weta. Dostajemy wtedy taki wynik, ponieważ

$$\prod_i v_i = \prod_i g^{x_i y_i} = g^{\sum_i x_i y_i} = g^0 = 1$$

W przeciwnym wypadku, jeśli

$$\prod_i v_i \neq 1,$$

to przynajmniej jedna osoba nie zgodziła się. Wszystkie oddane głosy były anonimowe, nie ma możliwości odtworzenia przesłanych decyzji.

Źródła

[1] Katarzyna Burda. *Prawie jak z nut*. 2020. URL: https://www.newsweek.pl/wiedza/nauka/choroba-klamstwa-naukowcy-zbadali-dlaczego-klamiemy/nfg2fjl (visited on 06/19/2020).

[2] Feng Hao and Piotr Zieliński. "A 2-Round Anonymous Veto Protocol". In: *Security Protocols*. Ed. by Bruce Christianson et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 202–211. ISBN: 978-3-642-04904-0.