Rozdział

Symulacyjne badanie splątania w protokołach kryptograficznych[1]

Piotr GAWRON
Instytut Informatyki Teoretycznej i Stosowanej PAN

gawron@iitis.gliwice.pl

Jarosław MISZCZAK
Instytut Informatyki Teoretycznej i Stosowanej PAN

miszczak@iitis.gliwice.pl

Streszczenie

W rozdziale tym poruszone jest zagadnienie testowania protokołów kryptograficznych opartych na wykorzystaniu stanów splątanych. Celem opracowania jest zaprezentowanie możliwości pakietu quantum-octave do analizy wydajności protokołów w zależności od parametrów kanałów kwantowych. Pierwsza część pracy zawiera omówienie oraz analizę protokołu uwierzytelniania qubitów. W drugiej części zaprezentowane są wyniki symulacji jednego z protokołów przeprowadzone z wykorzystaniem pakiety quantum-octave autorstwa autorów.

1. Motywacja

Najlepiej rozwiniętą dotychczas dziedziną zastosowania kwantowego przesłania informacji jest kwantowa dystrybucja klucza (ang. *quantum key distribution*) wykorzystująca destruktywność pomiaru kwantowego. Zaproponowano wiele protokołów pozwalających dwóm komunikującym się stroną na ustalenie pewnego ciągu bitów (klucza) w sposób całkowicie bezpieczny (patrz: [8] i pozycje bibliografii tam wymienione). Niektóre z tych propozycji teoretycznych doczekało się realizacji praktycznej,

[1] Praca wykonana w ramach projektu KBN nr 7 T11C 017 21.

czego dowodem są dostępne w handlu komercyjne systemy bazujące na kwantowej dystrybucji klucza.

Ponieważ splątanie jest uważane za cechę charakterystyczną dla mechaniki kwantowej naturalne jest poszukiwanie rozwiązań kryptograficznych związanych z jego wykorzystaniem. Pierwszą propozycją zastosowania kanałów wykorzystujących stany splątane był protokół A. Ekerta ustalania klucza [10]. Nowe rozwiązania w dziedzinie kryptografii kwantowej coraz częściej wykorzystują stany splątane do wykonywania zadań takich jak szyfrowanie czy uwierzytelnianie wiadomości [1,4,5,7,11,13].

W pierwszej części rozdziału przestawione są rozwiązania wykorzystujących stany splątane w kryptografii: protokół Ekerta oraz protokół pozwalający na uwierzytelnianie wiadomości kwantowych. Następnie przedstawione są główne funkcji pakietu obliczeń kwantowych quantum-octave, które mogą być wykorzystane do analizy stanów splątanych. Ostatnia część rozdziału zawiera wyniki symulacji przeprowadzonych z użyciem omawianego pakietu. Pozwalają one na ocenę efektywność omawianych protokołów oraz na zaobserwowanie zależności pomiędzy bezpieczeństwem a obecnością splątania w układzie.

2. Kryptografia bazująca na splątaniu

Najprostsza metoda przesyłania informacji przy wykorzystaniu splątania opiera się na schemacie teleportacji kwantowej oraz gęstego kodowania [18]. Teleportacja pozwala na przesyłanie wiadomości kwantowej zapisanej w postaci stanu układu, natomiast gęste kodowanie umożliwia zwiększenie przepustowości kanału dzięki wymianie informacji kwantowej. Ponieważ schemat teleportacji można rozszerzyć [14,15] na wiele qubitów możliwe jest przesyłanie w ten sposób dłuższych wiadomości.

Obecnie największe nadzieje na rozwój informatyki kwantowej wiąże się z wykorzystaniem takich schematów w układach operujących na fotonach. Jednak poważnym ograniczeniem jest trudność uzyskania i utrzymania stanów splątanych w układach wieloqubitowych.

Wykorzystując stan kwantowy możemy zakodować dowolną wiadomość klasyczną przyporządkowując stanom $|0\rangle$ oraz $|1\rangle$ wartość 0 oraz 1. W ten sposób przedstawiony poniżej protokół podpisywania qubitów pozwala na podpisywanie także wiadomości klasycznych.

2.1. Protokół Ekerta ustalania klucza

Zaproponowany w [10] przez Artura Ekerta protokół ustalania klucza bazuje na nielokalności występującej w mechanice kwantowej.

Dwie strony które chcą ustalić losowy ciąg bitów generują maksymalnie splątane stany Bella:

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle\right) \tag{1}$$

Każda ze stron może wykonać pomiar na posiadanym przez siebie części układu. Po zakończeniu pomiarów obie strony ogłaszają jakie pomiary zostały wykonane. Jeżeli obie strony dokonały pomiaru tej samej obserwabli (wielkości mierzalnej), uzyskane wyniki są idealnie skorelowane.

Jednocześnie jakikolwiek podsłuch może zostać wykryty z dużym prawdopodobieństwem gdyż powoduje on zaburzenie korelacji występujących w układzie. Pozwala to na ustalenie bezpiecznego klucza klucza sesji, który może zostać wykorzystany do transmisji danych klasycznych.

2.2. Uwierzytelnianie qubitów

Jednym zadań rozpatrywanych w klasycznej kryptografii jest problem uwierzytelniania wiadomości. Nadawca ma za zadanie wysłanie wiadomości tak by odbiorca był przekonany, co do jej pochodzenia i aby niemożliwe było podszycie się pod nadawcę osoby postronnej. W klasycznej kryptografii istnieją dwie techniki pozwalające na rozwiązanie tego problemu [19]: podpisy cyfrowe oraz ciągi uwierzytelniania wiadomości (ang. *message authentication signature*).

Załóżmy, że Alicja chce wysłać do Boba wiadomość w ten sposób by był on przekonany co do adresata. Niech wiadomością tą będzie stan jednego qubitu opisany macierzą ρ_m o wymiarach 2 na 2. Aby wiadomość została uwierzytelniona Alicja dokłada do stanu ρ_m stan ρ_t służący do przechowywania podpisu. Dla uproszczenia załóżmy, że ρ_t jest także stanem jednego qubitu. Podpisana wiadomość jest opisana przez macierz $\rho_m \otimes \rho_t$. Aby uwierzytelnianie było możliwe konieczne jest prze podjęciem komunikacji wymienienie jakiejś wiadomości. W przypadku uwierzytelniania kwantowego Alicja i Bob muszą być w posiadaniu jednego z qubitów układu w stanie splatanym. Każda ze stron może wykonywać operacje jedynie na swojej części stanu splątanego.

Ważnym ograniczeniem w przypadku uwierzytelniania kwantowego jest wymaganie, aby stan układu przeznaczonego na podpis był stanem czystym, $\rho_t = |0\rangle\langle 0|$. Całość układu (wiadomość podpisywana + podpis + klucz) opisana jest macierzą gęstości

$$|\alpha\rangle\langle\alpha| \otimes \rho_m \otimes |0\rangle\langle 0|, \tag{2}$$

która opisuje stan układu czterech qubitów.

W celu podpisania wiadomości wykonywane są kolejno następujące czynności. W pierwszym kroku Alicja wykonuje operację mającą na celu wprowadzenie splątania pomiędzy kluczem a pozostałą częścią układu. Alicja może wykonywać jedynie operacje lokalne na swojej części pary $|\psi^-\rangle$, zatem macierz kodowania można zapisać jako

$$E_A = |0\rangle\langle 0|_A \otimes I + |0\rangle\langle 0|_A \otimes U_e. \tag{3}$$

Macierz E_A jest przykładem macierzy kontrolowanej – w szczególności Alicja może wykonać operację kontrolowanej negacji. Operacja kodowania jest odpowiedzialna za wprowadzenie splątania pomiędzy układem a kluczem.

Macierz U_e jest dowolna, znana publicznie i jej wybór zależy od specyfikacji protokołu. Znana jest ona zatem zarówno odbiorcy jak i potencjalnemu przeciwnikowi.

W dugim kroku Bob wykonuje operację dekodowania wiadomości. Odbywa się to poprzez wykonanie operacji $D_B = |0\rangle\langle 0|_B \otimes U_e^+ + |0\rangle\langle 0|_B \otimes I$. Macierz U_e^+ to macierz sprzężona do U_e^+, natomiast I reprezentuje operację identyczności.

Następnie wykonywana jest operacja śladu częściowego [18] względem części układu przechowującej klucz oraz podpis. Wiadomość zdekodowana to

$$\rho_m^d = \mathrm{tr}_{AT}(I \otimes D_B \rho_{m+T+key}^e), \tag{4}$$

gdzie $\rho_{m+T+key}^e$ to macierz stanu układu po zakodowaniu, a tr_{AT} oznacza operację śladu częściowego po podukładzie Alicji oraz podpisie. Dodatkowo Bob sprawdza stan podpisu. Jeżeli podpis jest w stanie $|1\rangle$ akceptuje wiadomość – w innym przypadku ją odrzuca.

3. Badania symulacyjne

3.1. Pakiet quantum-octave

Wykorzystanie układy kwantowych do przetwarzania i przesyłania informacji wymaga odpowiedniego ich odizolowania od wpływu środowiska. W praktyce nieuniknione jest pojawienie się błędów. W związku z koniecznością uwzględnienia tego w trakcie symulacji konieczna jest możliwość operowania na mieszankach stanów.

Autorzy zaimplementowali system obliczeń kwantowych w postaci pakietu quantum-octave. Jest to zbiór funkcji dla programu GNU Octave [20]. Większość z nich jest zaimplementowana w języku skryptowym natomiast funkcje operacji na bitach są zaimplementowane w C++ przy wykorzystaniu biblioteki `liboctave`. Oprogramowanie było testowane z wersja środowiska o numerze 2.1.57.

Więcej szczegółów można znaleźć w [21, 23]. Poniżej przedstawimy jedynie najważniejsze cechy pakietu quantum-octave, głównie związane z tematem bieżącej pracy.

3.2. Analiza splątania w quantum-octave

Analiza splątania możliwa jest dzięki implementacji miar splątania. Dostępne są funkcje:

- Entropy - entropia von Neumanna,
- Negativity - ujemność stanu,
- LogNegativity - logarytmiczna ujemność stanu,
- IsPPT - sprawdzenie czy dana macierz gęstości jest dodatnio określona po wykonaniu operacji transpozycji częściowej.

W przypadku stanów czystych układu dwuskładnikowego istnieje jednoznaczna miara splątania - entropia von Neumana stanu zredukowanego [6,16].

Wyliczenie tej miary dla stanu polega na, wykonaniu operacji śladu częściowego a następnie obliczeniu entropii von Neumana dla otrzymanego stanu.

W przypadku stanów mieszanych istnieje kilka propozycji miar splątania, lecz większość z nich nie pozwala na wykonanie obliczeń numerycznych. Szczególnie użyteczne jest ujemność [22]. Jest to miara, która wskazuje, w jakim stopniu dany stan nie spełnia kryterium Peres-Horodeckiego [6,17]. Funkcje zaimplementowane w pakiecie quantum-octave pozwalają na obliczenie stopnia splątania stanów układu dwuskładnikowego.

3.3. Funkcje implementujące protokół uwierzytelniania

Jako przykład wykorzystania możliwości pakietu quantum-octave została zaimplementowana funkcje

```
TagQubit (state, gate)
```

oraz

```
CheckTag (state, gate)
```

pozwalające na podpisanie wiadomości złożonej z jednego qubitu za pomocą jednego qubitu przy wykorzystaniu stanu maksymalnie splątanego. Implementuje ona protokół zaimplementowany przez M. Curty *el al.* w [7].

Pobiera ona jako parametr operację unitarną, która służy do stworzenia operacji kontrolowanej wykonywanej przez Alice.

Dodatkowo stworzone zostały funkcje

```
TagQubitWithKey (message, gate, key)
```

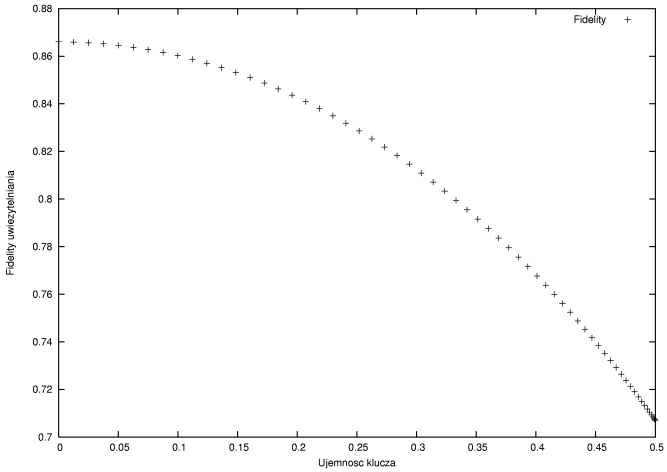
oraz

```
TagQubitWithBitError (message, gate, bit_error)
```

odpowiadające scenariuszom manipulacji klucza oraz zakłócenia macierzy pojawiających się w związku z wykonaniem algorytmu.

3.3.1. Manipulacja kluczem

Do przeprowadzenia uwierzytelniania potrzebna jest para qubitów w stanie maksymalnie splątanym. Jednakże otrzymanie pary maksymalnie splatanej nie zawsze jest zadaniem łatwym.



Rys. 1. Zależność wierności przekazu od ujemności stanu użytego jako klucz.

Przymnijmy, iż stan użyty jako klucz jest generowany w następujący sposób:

```
# stan wejściowy
s0 = State(kron(Id,H)*Ket([0,1]))
# generowanie bramek
for k=1:200
   gate(k).mtx = ControlledGate (2, RotY(2*k*pi/200), [2], [1]);
endfor
# tworzenie kluczy użytych do uwierzytelniania
for k=1:200
   key(k).mtx = Evolve(gate(k).mtx,s0);
endfor
```

Otrzymujemy w ten sposób ciąg kluczy zmienionych w sposób zależny od wielkości przesunięcia fazowego Rysunek bramce

```
gate(k).mtx = ControlledGate (2, RotY(2*k*pi/200), [2], [1]);
```

Na rys. 1 przedstawiona jest zależność wierności przekazu od ujemności stanu wygenerowanego w podany sposób użytego jako klucz.Wynika z tego, iż wykorzystanie dowolnych stanów w charakterze klucza powoduje zmniejszenie wydajności transmisji.

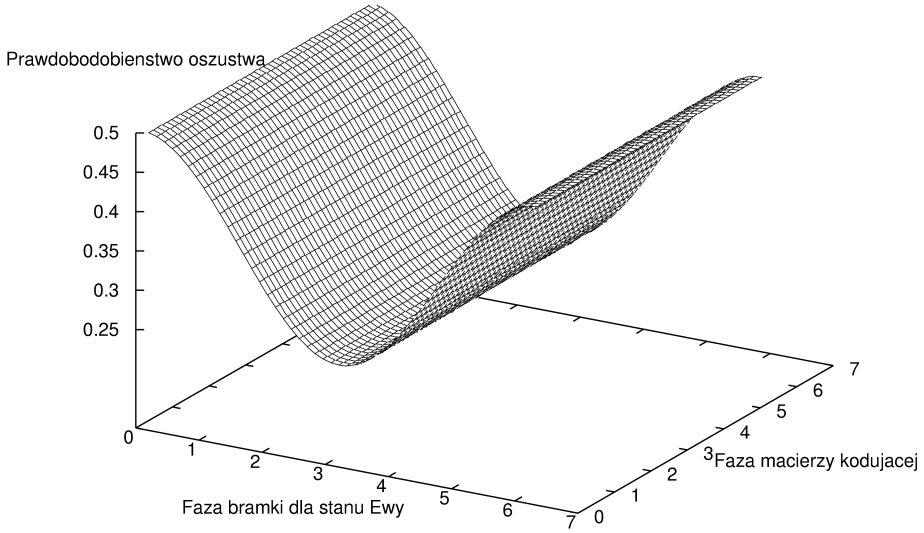
3.3.2. Atak

Strona obserwująca komunikację pomiędzy Alicją i Bobem (Ewa) może przechwycić transmitowaną wiadomość i spróbować oszukać Boba, co do swojej tożsamości. Atak tego typu nazywany jest atakiem typu *no-message attack*.

Załóżmy, iż Ewa generuje losowy stan mieszany, koduje go i wysyła do Boba. Przyjmijmy, iż stan Ewy jest maksymalną mieszanką zmodyfikowaną macierzą σ_X

$$\rho_e = RotX(k) \otimes RotX(l)(I \otimes I) \otimes |\psi^+\rangle\langle\psi^+|, \tag{5}$$

gdzie l oraz k to parametry.



Rys. 2. Zależność prawdopodobieństwa powodzenia ataku od fazy bramki której użyje ona do przygotowania stanu

Na rys. 2 przedstawione jest prawdopodobieństwo, z jakim Ewa oszuka Boba w zależności od przyjętych parametrów.

Macierz kodująca wykorzystana do podpisania wiadomości w scenariuszu ataku miała postać

```
ControlledGate (2, Phase(eve_phase(l)),[1],[2])
```

natomiast wiadomość której użyła Ewa do podszycia się

```
kron (RotY(gate_phase(k)), RotY(gate_phase(k))) * MaxMix(2)
```

czyli Ewa wygenerowała swoją wiadomość ze stanu maksymalnie mieszanego za pomocą pewnej bramki działającej na pojedyncze qubitu. Parametry k i l przebiegały liczby naturalne od 1 do 100.

Widać, że bezpieczeństwo algorytmu nie zależy od macierzy kodującej – jest to cecha, której należy oczekiwać od każdego algorytmu kryptograficznego. Z drugiej strony widać, iż Ewa nie osiągnie większego niż 0.5 prawdopodobieństwa podszycia się pod Alicję i oszukania Boba.

4. Wnioski

Splątanie jest kluczowym zasobem koniecznym do prawidłowego działania kwantowych protokołów szyfrowania. Schemat uwierzytelniania podobny do przedstawionego został zaproponowany także w odniesieniu do wiadomości klasycznych [8], zatem możliwe jest wykorzystanie go także do uwierzytelniania transmisji przeprowadzany w sieciach dysponujących dwoma rodzajami kanałów – kwantowych i klasycznych.

Przedstawione wyniki pokazują przydatność pakiety quantum-octave do realizacji prostych symulacji protokołów kwantowych. Ograniczenia na wielkość symulowanego układu wynikają głównie z wydajność pakietu. Jednakże w przypadku układów niskowymiarowych pakiet daje duże możliwości i pozwala na łatwa implementację algorytmów i protokołów.

LITERATURA

1. Barnum H.: Quantum message authentication codes, arXiv:quant-ph/0103123, 2001.
2. Bennett C. H., Bessette F., Brassard G., Salvail L., Smolin J.: Experimental quantum cryptography, Journal of Cryptology, vol. 5, no. 1, 1992, pp. 3-28.
3. Bouwmeester D. Ekert A., Zeilinger A.: The physics of quantmu computation, Springer, 2001.
4. Brassard G.: A bibliography of quantum cryptography, http://www.cs.mcgill.ca/~crepeau/CRYPTO/Biblio-QC.html, 1998.
5. Brassard G., Crépeau C., Jozsa R., Langlois D.: A quantum bit commitment scheme provably unbreakable by both parties, Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, pp. 362-371, 1993.
6. Bruss, D.: Characterizing entanglement, J. Math. Phys, vol. 43, 4327, 2002.
7. Curty M., Santos D. J., Perez E., Garcia-Fernandez P.: Qubit authentication, Physica A, 314, pp.130-139, 2002.
8. Curty M., Santos D. J.: Quantum authentication of classical messages, arXiv:quant-ph/0012077, 2000.
9. Dumais P., Touchette H.: QuCalc - Quantum Calculator, http://crypto.cs.mcgill.ca/QuCalc/, 2004.
10. Ekert A.: Quantum cryptography based on Bell theorem, Phys. Rev. Lett. 67, 661–663, 1991.
11. Gawron P., Grochla K.: Kwantowa dystrybucja klucza w sieciach optycznych, Studia Informatica, vol. 24, Nr. 2B (54), 2003.
12. Gawron P., J. A. Miszczak J. A.: Simulations of quantum systems evolution with quantum-octave package, zaakceptowane do druku w: Annales Informatica UMCS, 2004.
13. Gottesman D., Chuang I.: Quantum digital signatures, arXiv:quant-ph/0105032, 2001.
14. Grudka A.: Quantum teleportation between multiparties, arXiv:quant-ph/0303112, 2003.
15. Grudka A., Wojcik A.: Multiparty d-dimensional quantum information splitting , Acta Phys. Pol. A 102, 703, 2002.
16. Horodecki M.: Entanglement measures, Quantum Information and Computation, Vol. 1, No. 1, pp. 3-26, 2001.
17. Horodecki, M. Horodecki P., Horodecki R.: Separability of mixed states: necessary and sufficient conditions, Phys. Lett. A, vol. 223 (1-2), 1-131, 1996.
18. Preskill J.: Lecture Notes in Quantum Computation, http://www.theory.caltech.edu/people/preskill/ph229/, 2001.
19. Schneier B.: Kryptografia dla praktyków, Wydawnictwa Naukowo-Techniczne, Warszawa, 1995.
20. Strona projektu GNU Octave, http://www.octave.org/, 2004.
21. Strona projektu quantum-octave, http://quantum-octave.sourceforge.net/, 2004.

22. Vidal G., Werner R. F.: Computable measure of entanglement, Phys. Rev. A, vol. 65, 032314, 2002.