Entanglement-assisted random access codes

Marcin Pawłowski and Marek Żukowski
Institute of Theoretical Physics and Astrophysics, University of Gdańsk, PL-80-952 Gdańsk, Poland
(Received 30 June 2009; published 27 April 2010)

An (n,m,p) random access code (RAC) makes it possible to encode n bits in an m-bit message in such a way that a receiver of the message can guess any of the original n bits with probability p greater than $\frac{1}{2}$. In quantum RACs (QRACs), one transmits n qubits. The full set of primitive entanglement-assisted random access codes (EARACs) is introduced, in which parties are allowed to share a two-qubit singlet. It is shown that via a concatenation of these, one can build for any n an $(n,1,p)$ EARAC. QRACs for $n > 3$ exist only if parties also share classical randomness. We show that EARACs outperform the best of known QRACs not only in the success probabilities but also in the amount of communication needed in the preparatory stage of the protocol. Upper bounds on the performance of EARACs are given and shown to limit also QRACs.

DOI: 10.1103/PhysRevA.81.042326 PACS number(s): 03.67.Ac, 03.67.Bg, 03.67.Dd

I. INTRODUCTION

In many communication-related tasks quantum protocols are superior to classical ones. In cryptography, secret sharing, and communication complexity two particular approaches to "quantization" of classical tasks are used. The parties use quantum communication instead of classical communication, or the communication stays classical but the parties are allowed to share some entanglement. Examples for both approaches include [1] and [2] in cryptography, [3] and [4] in secret sharing, and [5] and [6] in communication complexity.

Quantum random access codes (QRACs), since their introduction in [7], have been studied only as protocols with quantum communication. Recently, an experimental realization of QRACs has been demonstrated [8]. As RACs are extremely useful in information processing tasks (e.g., network coding [9]), it is important to search for optimal codes. We address the problem of whether protocols that involve classical communication and entanglement lead to better codes. The answer is positive.

II. PRIMITIVES

Every RAC is described by three numbers n, m, and p, where n is the number of bits $(a_0,a_1,\ldots,a_{n-1})$ that are known only by the first party (Alice) and m is the number of bits she sends to the second party (Bob), $m < n$. The code is optimized in such a way that for every bit a_i of Alice, the probability that Bob correctly guesses this bit is at least p. Such code is denoted by (n,m,p) or, in the case when the probability is not specified but is strictly greater than $\frac{1}{2}$, by (n,m). In the case of a QRAC the communicated bits are replaced by qubits. An entanglement-assisted random access code (EARAC) is a code in which the m communicated bits are classical; however, the parties are additionally allowed to use shared entangled states during their coding and decoding procedures. For a specified m the only figure of merit to be studied is p as a function of n.[1] Other parameters, like the amount of entanglement necessary for the code, are usually not taken into account. However, if we were to introduce some additional efficiency factor that quantifies the amount of communication required in the preparatory stage of the protocol, EARACs have the upper hand also in this area. To be able to compare different types of resources, we first need to equate a single bit of shared randomness (SR) with a single e-bit. This is justified, since both of these elementary resources can be generated by a transfer of a single qubit from one party to another. Now we note that for $(n,1)$ code, the EARACs presented in this article require at most $n-1$ e-bits, while (for $n > 3$) QRACs require at least n bits of SR [11].

Let us start with pinpointing the two primitive EARACs, which are $\mathbf{(}2,1,\frac{1}{2}(1+\frac{1}{\sqrt{2}})\mathbf{)}$ and $\mathbf{(}3,1,\frac{1}{2}(1+\frac{1}{\sqrt{3}})\mathbf{)}$ ones, achievable with a shared two-qubit singlet and without any shared classical randomness. In the first case, denoted as $E^{[2]}$, Alice encodes her two bits, a_0 and a_1, by making a measurement in a basis dependent on the value of $a_0 \oplus a_1$. The two bases of Alice $A_{a_0\oplus a_1}$ are specified by the Bloch vector pairs $A_a = \{\pm\frac{1}{\sqrt{2}}\mathbf{(}1,(-1)^a,0\mathbf{)}\}$, where $a = 0,1$. Alice's outcome A is denoted as 0 if the measurement results in a collapse onto the first "+" state of the basis and 1 if onto the second one. Whenever Bob wants to learn a bth bit of Alice, he chooses to measure in the basis B_b, defined as

$$B_0 = \{\pm(1,0,0)\}, \quad B_1 = \{\pm(0,1,0)\}. \tag{1}$$

Bob's outcome B is ascribed 0 for the $-$ vectors and 1 for $+$ ones. Alice sends to Bob a message $M = a_0 \oplus A$. With this he can decode the desired bit with a high probability. As $P(A \oplus B = 0) = \frac{1}{2}(1+\vec{a}\cdot\vec{b})$, where $\vec{a}$ and $\vec{b}$ are the Bloch vectors of the local settings specified by the $+$ vectors of the bases, which in turn implies that $P(A \oplus B = ab) = \frac{1}{2}(1+\frac{1}{\sqrt{2}})$, it is easy to see that

$$M \oplus B = a_b, \tag{2}$$

with a probability $p = \frac{1}{2}(1+\frac{1}{\sqrt{2}}) \approx 0.85$. This code may be considered as a version of an oblivious transfer protocol presented in [12] with the singlet state being an imperfect realization of a PR box [13].

[1] We study the case of $m = 1$. A generalization to $m > 1$ will be presented elsewhere [10].

To construct a $\left(3,1,\frac{1}{2}(1+\frac{1}{\sqrt{3}})\right)$ EARAC, denoted here as $E^{[3]}$, a similar procedure can be used. The only difference is in the bases that the parties choose. For Alice there are four bases and the choice is again implied by her input bits

$$\begin{aligned} a_0 = a_1 = a_2 &\Rightarrow A_0 = \left\{\pm\frac{1}{\sqrt{3}}(1,1,1)\right\}, \\ a_0 = a_1 \neq a_2 &\Rightarrow A_1 = \left\{\pm\frac{1}{\sqrt{3}}(1,1,-1)\right\}, \\ a_0 \neq a_1 = a_2 &\Rightarrow A_2 = \left\{\pm\frac{1}{\sqrt{3}}(1,-1,-1)\right\}, \\ a_0 \neq a_1 \neq a_2 &\Rightarrow A_3 = \left\{\pm\frac{1}{\sqrt{3}}(1,-1,1)\right\}. \end{aligned} \tag{3}$$

The outcome and the message are defined exactly in the same way as in the previous (2,1) code. Bob again uses mutually unbiased bases B_0 and B_1 from (1) but this time, if he aims at the third bit, he chooses $B_2 = \{\pm(0,0,1)\}$. It is easy to see that in this case the probability that $M \oplus B = a_b$ is $p = \frac{1}{2}(1+\frac{1}{\sqrt{3}}) \approx 0.79$. To our knowledge this protocol has not been studied so far.

The success probabilities for these codes exactly match the ones for known QRACs [7,9]. This is not surprising. The basis vectors of Alice in EARAC correspond exactly to her quantum code words in QRAC, and Bob's measurements are the same.

In [9] it was shown that a (4,1) QRAC (without SR) does not exist [this also holds for any $(n,1)$ QRAC with $n > 4$]. Nevertheless, as we show, for any n, one has an $(n,1)$ EARAC. To this end we present concatenated EARACs formed out of the primitive ones singled out previously.

III. CONCATENATION

Since the procedure has classical inputs and outputs at every point, it can be concatenated. Consider the following simplest case. Alice is given four bits a_0, a_1, a_2, and a_3, while Bob might be interested in a bit number $b = 0,\ldots,3$. They agree that Bob would use a binary expansion of b, given by a bit sequence b_1b_0 defined via $\sum_{i=0}^{1} b_i 2^i$, where $b_i = 0,1$. Alice encodes the first two bits of hers by performing a measurement in a basis $A_{a_0\oplus a_1}$ on a (2,1) EARAC called here $E(b_1 = 0)$. Denote the measurement result by $A^{(0)}$. Her output value is $M_0 = a_0 \oplus A^{(0)}$. She also encodes the other two bits in a similar procedure with yet another (2,1) EARAC, denoted as $E(b_1 = 1)$. The basis is now $A_{a_2\oplus a_3}$. The output is $M_1 = a_2 \oplus A^{(1)}$, where $A^{(1)}$ is her measurement result for $E(b_1 = 1)$. The value of M_1 along with M_0 are treated as her input bits for a third (2,1) EARAC, denoted by E', which fixes the measurement basis as $A_{M_0\oplus M_1}$. Let us denote the result by A'. Only the output bit of this final EARAC, $M = M_0 \oplus A'$, is then sent to Bob. What Alice does is an encoding of a_0 and a_1 into M_0 and a_2 and a_3 into M_1, while M_0 and M_1 are next encoded into M. When Bob gets M and performs a measurement on E' in the basis B_{b_1} yielding a result denoted as B', he is able to decode M_{b_1} if he also performs a measurement in basis B_{b_0} on $E(b_1)$. The results for $E(b_1 = x)$ we denote by $B^{(x)}$. The value of his guess is then $a_{b_1b_0} = M \oplus B^{(b_1)} \oplus B'$. Bob makes measurements on only two of three singlets at his disposal, but which of them are measured depends on his choice of bit he is interested in. Once he chooses to measure E' in the basis $B_{b_1=x}$, this can be accompanied by a measurement on $E(b_1 = x)$, a measurement on $E(b_l \neq x)$ is useless. A larger number of bits encoded comes at the price of lower probability of success. Bob will guess the target bit correctly if both of their EARAC devices give the correct value or if both of them are wrong. Therefore, the code just introduced is a $(4,1,\frac{3}{4})$ EARAC (see later in this article).

More generally, if Alice and Bob know how to devise a $(k,1,p_k)$ EARAC, denoted here $E^{[k]}$, they can use two such procedures followed by a (2,1) one to construct a $(2k,1,p_{2k})$ code. Alice simply encodes first k bits into one the first $E^{[k]}$ using the coding procedure from $(k,1,p_k)$ code and she repeats the procedure for the remaining k bits with the second $E^{[k]}$. That allows her to compress $2k$ bits into two, which she can encode using a (2,1) primitive EARAC into a single-bit message, which is sent to Bob. Again, Bob needs two failures or two successes of their EARAC devices to get the correct value of the bit he is interested in. This yields the probability of success

$$p_{2k} = p_k\frac{1}{2}\left(1+\frac{1}{\sqrt{2}}\right) + (1-p_k)\frac{1}{2}\left(1-\frac{1}{\sqrt{2}}\right), \tag{4}$$

which if one puts $p_k = \frac{1}{2}(1+d)$ reads $p_{2k} = \frac{1}{2}(1+d\frac{1}{\sqrt{2}})$ By induction, since $p_1 = \frac{1}{2}(1+\frac{1}{\sqrt{2}})$, this procedure makes it possible to generate a $\left(2^k,1,\frac{1}{2}(1+2^{-\frac{1}{2}k})\right)$ EARAC.

To generate an $(n,1)$ EARAC for $n > 3$, one must use the primitive ones, $E^{[2]}$ and $E^{[3]}$, in a concatenated scheme (see Fig. 1). For example, for $n = 5$ Alice will encode first three bits using a $E^{[3]}$ and the remaining two with $E^{[2]}$. Then she uses one more $E^{[2]}$ to get one bit of message that she sends to Bob. In the general case, if Alice gets n bits she can divide them into n_2 groups of 2 and n_3 groups of 3. If the reminder of the division of n by 3 is r then $n_2 = 2r \bmod 3$ and $n_3 = n - 2n_2$. This gives her $n' = \frac{n_3}{3} + \frac{n_2}{2} \geqslant n$ bits, with which she will perform a similar procedure until she is left with only 1 bit that she can communicate to Bob (of course, they ignore the possible additional bit by fixing it to 0).

To calculate the success probability of guessing a given bit, one has to trace the way from the initial bit to the final coded message in the schematic representation of the code (see Fig. 1 for example). Since the success probability for a $E^{[3]}$ is lower than for $E^{[2]}$, the success probability of the code will depend on which primitives were used in the concatenation. The guess is successful not only when the guess is correct for every step of the concatenation, but also if the number of errors is even. The probability of making an even number of errors when using $E^{[2]}$ k times is

$$\begin{aligned} p_{2e}(k) &= \sum_{i=0}^{\lfloor\frac{k}{2}\rfloor}\binom{k}{2i}\left[\frac{1}{2}\left(1+\frac{1}{\sqrt{2}}\right)\right]^{k-2i}\left[\frac{1}{2}\left(1-\frac{1}{\sqrt{2}}\right)\right]^{2i} \\ &= \frac{1}{2}(1+2^{-\frac{1}{2}k}). \end{aligned} \tag{5}$$

For a $E^{[3]}$ it is

$$p_{3e}(k) = \tfrac{1}{2}(1+3^{-\frac{1}{2}k}). \tag{6}$$

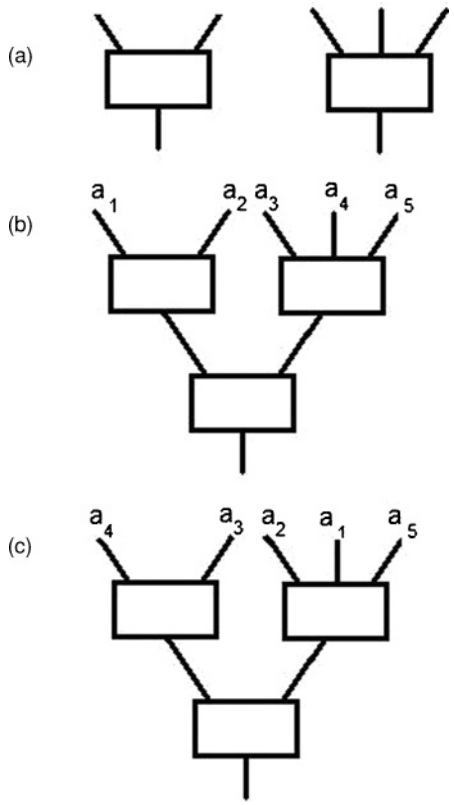

FIG. 1. (a) Schematic representation of the two primitives: (2,1) and (3,1) EARAC. (b) Example of concatenation: (5,1) EARAC. The probability of Bob guessing a_2 is higher than for a_3 since in the first case only (2,1) EARAC primitives are used. These probabilities are, respectively, $P_2 = \frac{1}{2}(1 + \frac{1}{\sqrt{4}})$ and $P_3 = \frac{1}{2}(1 + \frac{1}{\sqrt{6}})$. The success probability associated with the whole code is the lower one. (c) One possible permutation of input bits for (5,1) code. The choice of permutation is governed by the data from a shared random string. Having all possible permutations with the same probability results in an averaged version of the protocol with the success probability $p = \frac{2}{5}P_2 + \frac{3}{5}P_3$, which is higher than in the previous case.

For an odd number of errors, one has

$$p_{2o}(k) = \tfrac{1}{2}(1 - 2^{-\frac{1}{2}k}), \tag{7}$$

$$p_{3o}(k) = \tfrac{1}{2}(1 - 3^{-\frac{1}{2}k}). \tag{8}$$

That gives the overall probability of success for a bit encoded k times with (2,1) EARACs and j times with (3,1) EARACs:

$$p_{k,j} = p_{2e}(k)p_{3e}(j) + p_{2o}(k)p_{3o}(j) = \tfrac{1}{2}(1 + 2^{-\frac{1}{2}k}3^{-\frac{1}{2}j}). \tag{9}$$

Thus, for any encoding of this type the probability of guessing any bit is strictly greater than $\frac{1}{2}$. Therefore, in contrast to QRACs, an $(n,1)$ EARAC exists for any n, even if the parties do not make use of SR.

When parties are allowed to use SR, QRACs do exist for any n (see [11]), which is obvious, for anyone can devise nonconcatenated EARAC using in the same way SR as their counterpart QRACs. However, we will show that for concatenated EARAC with SR, the success probabilities are higher than for the best of known QRACs.

IV. SHARED RANDOMNESS

One of the possibilities of employing SR in an EARAC is via defining the place each bit of Alice a_i enters the coding procedure by the values of the random string. This makes it possible to average the success probabilities. Obviously, this cannot decrease the minimal probability, which defines the efficiency of the code. In fact, in most of the cases this efficiency is increased. For an $(n,1,p)$ EARAC with n_i bits having success probability p_i, where $n = \sum_i n_i$ and $p = \min p_i$, SR allows an upgrade to an $(n,1,p')$ EARAC, where $p' = \frac{1}{n}\sum_i n_i p_i$ (see Fig. 1).

An evident advantage of an EARAC with SR over a QRAC with SR from [11], which are the best ones known so far, is the simpler way of finding a construction. To create the EARAC one just needs to concatenate the two basic codes a sufficient number of times. In the case of a QRAC, numerical search procedures are used. The other advantage is that EARACs give higher probabilities of success for all $n > 3$ (see Table I).

V. UPPER AND LOWER BOUND

Calculating the exact success probability for $(n,1,p)$ EARAC is not difficult, but it is hard to put into a short formula. Nevertheless, it is possible in the special case when the success probabilities for all individual bits are equal, without a randomization procedure with the use of SR. If k and j again denote the number of times (2,1) and (3,1) are used for encoding a bit, $p_{k,j}$ will be the same for all bits if and only if for *all* bits k and j are the same. This implies that the preceding conditions hold for only $n = 2^k 3^j$. In such a case, via Eq. (9), the success probability is

$$p = p_n = \frac{1}{2}\left(1 + \frac{1}{\sqrt{n}}\right). \tag{10}$$

Equation (10) gives also an upper bound for the success probability of any EARAC for any n. Consider an

TABLE I. Success probabilities for $(n,1,p_{Q,n})$ QRAC with SR from [11] compared with the probabilities for $(n,1,p_{E,n})$ EARAC. The rightmost column displays the advantage of EARAC for every $n > 3$.

n	$p_{Q,n}$	$p_{E,n}$	$\Delta = p_{E,n} - p_{Q,n}$
2	$\frac{1}{2}(1 + \frac{1}{\sqrt{2}})$	$\frac{1}{2}(1 + \frac{1}{\sqrt{2}})$	0
3	$\frac{1}{2}(1 + \frac{1}{\sqrt{3}})$	$\frac{1}{2}(1 + \frac{1}{\sqrt{3}})$	0
4	0.741 48	$\frac{3}{4}$	0.008 52
5	0.713 58	$\frac{1}{20}(12 + \sqrt{6})$	0.008 89
6	0.694 05	$\frac{1}{2}(1 + \frac{1}{\sqrt{6}})$	0.010 07
7	0.678 64	$\frac{1}{21}(12 + \sqrt{6})$	0.009 43
8	0.666 63	$\frac{1}{80}(52 + \sqrt{6})$	0.013 99
9	0.656 89	$\frac{2}{3}$	0.009 78
10	0.648 20	$\frac{1}{20}(10 + \sqrt{2} + \sqrt{3})$	0.009 11
11	0.641 05	$\frac{1}{120}(60 + 3\sqrt{2} + 8\sqrt{3})$	0.009 78
12	0.634 87	$\frac{1}{2}(1 + \frac{1}{\sqrt{12}})$	0.009 47
15	0.620 36	$\frac{1}{60}(30 + 3\sqrt{2} + 2\sqrt{3})$	0.008 09

$(n^N,1)$ EARAC constructed by N-level concatenation of $\mathbf{(}n,1,\frac{1}{2}(1+\frac{1}{\sqrt{n}}+\epsilon)\mathbf{)}$ EARACs. Since Bob will guess every bit in the concatenated protocol with the same probability, the success probability is given by (10)

$$p_{n^N}=\frac{1}{2}\left[1+\left(\frac{1}{\sqrt{n}}+\epsilon\right)^N\right]. \tag{11}$$

To proceed further, we shall use the following technical result. In [14] it is shown that, in the general case on any information processing protocol, classical or quantum, if Alice holds a K-bit secret entirely random data string and sends *one* classical bit to Bob, then the bound on mutual information implies the following inequality:

$$K[1-h(p_K)]\leqslant 1, \tag{12}$$

where p_K denotes the probability that Bob guesses correctly any of her K bits, and h is Shannon binary entropy. If we substitute $K=n^N$ and use $1-h(\frac{1+y}{2})\geqslant\frac{y^2}{2\ln 2}$, we get

$$K[1-h(p_K)]\geqslant\frac{n^N\left(\frac{1}{\sqrt{n}}+\epsilon\right)^{2N}}{2\ln 2}=\frac{(1+2\sqrt{n}\epsilon+n\epsilon^2)^N}{2\ln 2}. \tag{13}$$

If $\epsilon>0$, then $1+2\sqrt{n}\epsilon+n\epsilon^2>1$ and there exists N large enough for which $(1+2\sqrt{n}\epsilon+n\epsilon^2)^N>2\ln 2$, which leads to violation of (12). That means that for any $(n,1,p_n)$ EARAC, one must have

$$p_n\leqslant\frac{1}{2}\left(1+\frac{1}{\sqrt{n}}\right). \tag{14}$$

Note that the bound (12) also holds for QRACs. To see this, notice that for any $(n,1,p)$ QRAC, there exists $(n,1,p)$ EARAC with the same p. It can be constructed in the following way. Alice and Bob share a singlet state. When Alice wants to encode her input $a_1,\ldots,a_n$ she calculates $\rho(a_1,\ldots,a_n)$, which would be the (pure) state that she would send to Bob if they were to use $(n,1,p)$ QRAC. She then measures her part of the singlet in the basis that includes $\rho(a_1,\ldots,a_n)$ as one of its eigenstates. Her measurement outcome tells her whether the part of the singlet at Bob's laboratory collapsed to $\rho(a_1,\ldots,a_n)$ or the orthogonal one. She then sends Bob 0 if it is this state and 1 if it is orthogonal. When Bob wants to find the value of the ith bit, he performs the measurement that he would in the case of QRAC. If he has received message 0 from Alice, he keeps his outcome; if 1, he flips it. That procedure gives him the same probability of successfully guessing any bit as in QRAC. Therefore, any bound on such EARACs is also valid for QRACs. Our proof is thus also a simpler version of the one for the bound for QRAC presented in [11]. The bound (14) is saturated whenever n is of the type 2^k3^j. Thus, in such cases the presented EARACs are optimal.

As has already been mentioned, the derivation of success probability for any given EARAC is straightforward, but it is difficult to give one formula for the general case. It is, however, possible to give a lower bound for optimal protocols. Notice that if n is not of the form 2^k3^j, the parties can always use an $(n_\geqslant,1)$ EARAC, where $n_\geqslant$ is the smallest integer greater than n and being of the form 2^k3^j. This leads to the lower bound

$$p_n\geqslant\frac{1}{2}\left(1+\frac{1}{\sqrt{n_\geqslant}}\right). \tag{15}$$

VI. CONCLUSION

We introduce EARACs and show them to be superior to the best known QRACs in the terms of both success probabilities and existence without SR. We also derive the bound for any RAC and show that for infinitely many n the EARAC is the best one. An open questions is whether the bound is saturated for all n, and if not, what are then the best possible codes.

ACKNOWLEDGMENTS

We thank T. Paterek for discussions. This work has been supported by EU programme QAP (No. 015848). It has been done at the National Quantum Information Centre of Gdansk (NQuantIC).

[1] C. H. Bennett and G. Brassard, in Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India (IEEE, New York, 1984).
[2] A. K. Ekert, Phys. Rev. Lett. **67**, 661 (1991).
[3] C. Schmid, P. Trojek, M. Bourennane, C. Kurtsiefer, M. Żukowski, and H. Weinfurter, Phys. Rev. Lett. **95**, 230505 (2005).
[4] M. Żukowski, A. Zeilinger, M. A. Horne, and H. Weinfurter, Acta Phys. Pol. **93**, 187 (1998); M. Hillery, V. Bužek, and A. Berthiaume, Phys. Rev. A **59**, 1829 (1999); R. Cleve, D. Gottesman, and H.-K. Lo, Phys. Rev. Lett. **83**, 648 (1999); A. Karlsson, M. Koashi, and N. Imoto, Phys. Rev. A **59**, 162 (1999).
[5] E. F. Galvao, Phys. Rev. A **65**, 012318 (2001); J. N. de Beaudrap, *ibid.* **69**, 022307 (2004); R. T. Horn, S. A. Babichev, K.-P. Marzlin, A. I. Lvovsky, and B. C. Sanders, Phys. Rev. Lett. **95**, 150502 (2005); P. Trojek, C. Schmid, M. Bourennane, Č. Brukner, M. Żukowski, and H. Weinfurter, Phys. Rev. A **72**, 050305(R) (2005).
[6] H. Buhrman, R. Cleve, and W. van Dam, SIAM J. Comput. **30**, 1829 (2001).
[7] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, J. ACM **49**, 496 (2002).
[8] R. W. Spekkens, D. H. Buzacott, A. J. Keehn, B. Toner, and G. J. Pryde, Phys. Rev. Lett. **102**, 010401 (2009).
[9] M. Hayashi, K. Iwama, H. Nishimura, R. Raymond, and S. Yamashita, New J. Phys. **8**, 129 (2006).
[10] M. Markiewicz and M. Pawłowski (unpublished).
[11] A. Ambainis, D. Leung, L. Mancinska, and M. Ozols, e-print arXiv:0810.2937.
[12] S. Wolf and J. Wullschleger, e-print arXiv:quant-ph/0502030.
[13] S. Popescu and D. Rohrlich, Found. Phys. **24**, 379 (1994).
[14] M. Pawłowski, T. Paterek, D. Kaszlikowski, V. Scarani, A. Winter, and M. Żukowski, Nature (London) **461**, 1101 (2009).