Ralph's BB84 QKD

Provably secure encryption takes us from EPR QKD to CSS QKD, finally, to BB84 QKD. This is Ralph's quantum encryption except rather than one bit a time the entire m-bit key is packaged in n-bit code. The protocol, a derivative of the protocols mentioned above (see Shor and Preskill), is as follows.

1. Ralph creates $4n$ plus random bits and $4n$ random bit string b.

2. For each bit, Ralph encodes a qubit in the $|0\rangle, |1\rangle$ (Z) basis or the $|+\rangle, |-\rangle$ (X) basis in accordance with the b vector such that 0 results in Z basis and 1 results in X basis encoding. If the first string is 0 then either $|0\rangle$ or $|+\rangle$ is encoded, alternatively, if the first string is 1 then either $|1\rangle$ or $|-\rangle$ is encoded (based on b).[1]

3. Ralph sends the resultant qubits to Alice.

4. Alice receives the qubits, announces their receipt, and measures them in the X or Z basis at random.

5. Ralph announces b.

6. Alice and Ralph discard any qubits measured in a different basis. With high probability there are at least $2n$ bits remaining. If not, abort the protocol and try again. Ralph randomly decides the $2n$ bits to retain, randomly selects which n bits are check bits, and announces the selection.

7. Alice and Ralph publicly compare their check digits. If more than t (the error correcting capacity of the code) disagree they abort the protocol. Ralph is left with the n bit vector x and Alice is left with $x + \varepsilon$.

8. Ralph randomly chooses $v_k \in C_1$ and announces $x - v_k$. Alice subtracts this from her vector leaving $v_k + \varepsilon$. She corrects this to v_k using the syndrome and bit flip operator X_j.

9. Alice and Ralph use the coset of $v_k + C_2$ in C_1 to obtain the shared m bit private key k (see the appendix for coset decoding).

Suggested:

1. Suppose there is no interference and the randomly generated code is [0100110] and $x = [0000000]$, do Alice and Ralph share the same key?

2. Repeat 1 where the randomly generated code is [0100111] and $x = [0000001]$.

3. Suppose everything is as in 2 except the first qubit is bit flipped during transmission so that $x + \varepsilon = [1000001]$. Repeat 1.

[1] This scheme replaces Hadamard operations in EPR and CSS QKD.

Appendix coset decoding

For the Hamming code, coset decoding involves adding all variations of the parity check matrix (in C_2) to all linear combinations of legal codewords in C_1 (altering only the parity check entries), then finding the leader of the coset to which the codeword belongs, and adding the leader to the codeword to derive the decoded word.

The 16 legal codewords for the Hamming [7,4] code (in standard form; note, this differs from the form discussed in Andonian and Brankovic) are as follows:

$$0000000, 0011110, 0101011, 0110101, 1000111, 1011001, 1101100, 1110010,$$
$$1111111, 1100001, 1010100, 1001010, 0111000, 0100110, 0010011, 0001101.$$

A convenient way to write the parity check additions (modulo 2) is

$$0000000, 0000101, 0000011, 0000111, 0000100, 0000010, 0000001, 0000110.$$

Form 8 cosets by adding the parity check to each of the 16 legal codewords leaving the initial code bit unchanged.[2] This strategy identifies the nearest legal codeword to the string. The coset leader is the string in the coset with fewest non-zeroes (indicated by brackets).

Coset 1 is the set of legal codewords

$$[0000000], 0011110, 0101011, 0110101, 1000111, 1011001, 1101100, 1110010,$$
$$1111111, 1100001, 1010100, 1001010, 0111000, 0100110, 0010011, 0001101.$$

Coset 2 is

$$0000101, 0011011, 0101110, 0110000, 1000010, 1011100, 1101001, 1110111,$$
$$1111010, 1100100, 1010001, 1001111, 0111101, 0100011, 0010110, [0001000].$$

Coset 3 is

$$0000011, 0011101, 0101000, 0110110, 1000100, 1011010, 1101111, 1110001,$$
$$1111100, 1100010, 1010111, 1001001, 0111011, 0100101, [0010000], 0001110.$$

Coset 4 is

$$0000111, 0011001, 0101100, 0110010, [1000000], 1011110, 1101011, 1110101,$$
$$1111000, 1100110, 1010011, 1001101, 0111111, 0100001, 0010100, 0001010.$$

Coset 5 is

$$[0000100], 0011010, 0101111, 0110001, 1000011, 1011101, 1101000, 1110110,$$
$$1111011, 1100101, 1010000, 1001110, 0111100, 0100010, 0010111, 0001001.$$

Coset 6 is

$$[0000010], 0011100, 0101001, 0110111, 1000101, 1011011, 1101110, 1110000,$$
$$1111101, 1100011, 1010110, 1001000, 0111010, 0100100, 0010001, 0001111.$$

[2] Notice $16 \times 8 = 2^7 = 128$ covers every binary permutation. Since this is a Hamming code all strings are within one change of a legal codeword.

Coset 7 is

$[0000001], 0011111, 0101010, 0110100, 1000110, 1011000, 1101101, 1110011, 1111110, 1100000, 1010101, 1001011, 0111001, 0100111, 0010010, 0001100.$

Coset 8 is

$0000110, 0011000, 0101101, 0110011, 1000001, 1011111, 1101010, 1110100, 1111001, 1100111, 1010010, 1001100, 0111110, [0100000], 0010101, 0001011.$

Suppose the code is 0001011. This string resides in coset 8 and the leader of coset 8 is 0100000. Therefore, the decoded word is 0001011+0100000 = 0101011. This is the third legal codeword.