Vulnerability of "A Novel Protocol-Authentication Algorithm Ruling out a Man-in-the-Middle Attack in Quantum Cryptography"

Aysajan Abidin and Jan-Åke Larsson

Abidin, A., Larsson, J., (2009), Vulnerability of "A Novel Protocol-Authentication Algorithm Ruling out a Man-in-the-Middle Attack in Quantum Cryptography", *International Journal of Quantum Information*, 7(5), 1047-1052. https://doi.org/10.1142/S0219749909005754

Original publication available at:
https://doi.org/10.1142/S0219749909005754

International Journal of Quantum Information
Vol. 7, No. 5 (2009) 1047–1052

World Scientific
www.worldscientific.com

VULNERABILITY OF "A NOVEL PROTOCOL-AUTHENTICATION ALGORITHM RULING OUT A MAN-IN-THE-MIDDLE ATTACK IN QUANTUM CRYPTOGRAPHY"

AYSAJAN ABIDIN* and JAN-ÅKE LARSSON†
Department of Mathematics, Linköping University,
581 83 Linköping, Sweden
**aiabu@mai.liu.se*
†*jalar@mai.liu.se*

Received 9 May 2009

In this paper, we review and comment on "A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography" [M. Peev *et al.*, *Int. J. Quant. Inf.* **3** (2005) 225]. In particular, we point out that the proposed primitive is not secure when used in a generic protocol, and needs additional authenticating properties of the surrounding quantum-cryptographic protocol.

Keywords: Quantum cryptography; quantum key distribution; authentication.

1. Introduction

Quantum cryptography — or, more accurately, quantum key distribution (QKD) — is an unconditionally secure key growing technique based on the principles of quantum mechanics. It is unconditionally secure because no quantum state can be copied or measured without disturbing it. However, the practical implementation of QKD protocols requires an immutable public channel. If the public channel is not immutable, the eavesdropper (Eve) can easily mount a man-in-the-middle (MITM) attack, since Eve is in control of both the quantum and the public channel. For the attack to be successful, Eve needs, among other things, to substitute the classical message from one legitimate user (Alice) to the other (Bob) without being noticed. To prohibit such an attack on QKD, proper message authentication is needed. Therefore, QKD is secure only if it is combined with an unconditionally secure message authentication scheme. In this paper, we will review a recently proposed authentication primitive[1] and point out that it is not secure when used in a generic QKD system. It has earlier been shown[2] that an attack is possible against the "privacy amplification" step in a QKD protocol using the proposed authentication, but the attack presented here is more serious and enables a full MITM attack

on the whole system, unless some additional part of the protocol has authenticating properties.

2. The Proposed Authentication Primitive

In Ref. 1, the authors propose an authentication primitive which aims to decrease the key consumption for the authentication purposes in QKD, and in turn, to improve the efficiency of the key growth in QKD. The algorithm works as follows. Let $\mathcal{M}$ be the set of all binary strings of length m (or the set of all messages of length m), and let $\mathcal{T}$ be the set of all binary strings of length n with $n < m$ (or the set of all tags of length n). A message m_{A} is first mapped from $\mathcal{M}$ to $\mathcal{Z}$, where $\mathcal{Z}$ is the set of all binary strings of length r with $n < r < m$, by a single publicly known hash function f so that $z_{\mathrm{A}} = f(m_{\mathrm{A}})$. And then z_{A} is mapped by a secret $h_k \in \mathcal{H}_{\mathcal{Z}}$ to a tag $t_{\mathrm{A}} = h_k(z_{\mathrm{A}})$, where $\mathcal{H}_{\mathcal{Z}} : \mathcal{Z} \mapsto \mathcal{T}$ is a Strongly Universal$_2$ (SU_2) family of hash functions[3] and the subscript k is the secret key needed to identify a hash function. The message–tag pair $m_{\mathrm{A}} + t_{\mathrm{A}}$ will be sent over the public channel. To authenticate the message $m_{\mathrm{A}} \in \mathcal{M}$, the legitimate receiver computes $h_k(f(m_{\mathrm{A}}))$ and compares it to t_{A}. If they are identical, then the message will be accepted as authentic, otherwise, it will be rejected. Since r is fixed independently of m, the key length required for authentication is constant regardless of the message length to be authenticated.

This authentication algorithm is claimed[1] to be secure with a probability ϵ of Eve being able to create the correct tag for her fake message. In Ref. 1, this is calculated as[a]

$$\epsilon = \epsilon_1 + \epsilon_2, \tag{1}$$

where $\epsilon_2 = 1/|\mathcal{T}|$, which is the probability of guessing the correct tag when an SU_2 hash function family is used, and ϵ_1 is the probability that the message m_{A} and Eve's modified message $m_{\mathrm{E}} (\neq m_{\mathrm{A}})$ yield the same value under the publicly known hash function f.

3. The Problem

This authentication primitive is such that whenever Eve's message m_{E} happens to coincide with Alice's message m_{A} under the publicly known hash function f, i.e. $f(m_{\mathrm{E}}) = f(m_{\mathrm{A}})$, Eve can just send $m_{\mathrm{E}} + t_{\mathrm{A}}$ since $t_{\mathrm{E}} = t_{\mathrm{A}}$. The problem here is that in Ref. 1, security is derived under the explicit assumption that Eve has a fixed message. The result holds, but in generic QKD, Eve is *not* restricted to one message m_{E}.

In a full MITM attack on a QKD protocol, Eve impersonates Bob to Alice and Alice to Bob during the quantum transmission process and the subsequent public

[a]Actually, $\epsilon \leq \epsilon_1 + \epsilon_2$; Eq. (1) is an upper bound rather than an equality.

discussions. We use BB84[4] with simple reconciliation and privacy amplification, and *immediate authentication* of each phase as our first example. This would consist of, in order: raw key generation; sifting and immediate authentication; one-way error correction and immediate authentication; one-way privacy amplification and authentication (see e.g. Ref. 5, Chap. 12). Eve receives and measures the qubits that Alice has sent to Bob, in her choice of basis. We note here that although QKD requires that Bob randomly selects the basis in which to measure the qubits, Eve can ignore this requirement. At the same time, she chooses a set of qubits in, again, not necessarily random states and sends these to Bob. After Bob receives and measures the qubits sent by Eve in a randomly selected basis, he sends an authenticated time stamp to Alice to end the quantum transmission phase. Now, Alice sends her message m_{A}, which contains the settings used for encoding/decoding on the quantum channel, along with the authentication tag t_{A}, to Bob. Eve intercepts the message–tag pair and calculates $f(m_{\mathrm{A}})$ and compares it with $f(m_{\mathrm{E}})$. In the rare event that they are equal, Eve can just send $m_{\mathrm{E}} + t_{\mathrm{A}}$ to Bob. Otherwise, she can change her message m_{E}, which contains the settings. Changing one of the settings, i.e. changing one bit of the message, will at most introduce one noisy bit into the sifted key. Even a few noisy bits will not have a noticeable effect in practical QKD systems because of the error correction used in the reconciliation step.

In this situation, if $f(m_{\mathrm{E}}) \neq f(m_{\mathrm{A}})$, Eve can search for a message m'_{E} with $d_{\mathrm{Hamming}}(m_{\mathrm{E}}, m'_{\mathrm{E}}) = 1$ (or "small") such that $f(m'_{\mathrm{E}}) = f(m_{\mathrm{A}})$. In other words, she tries to find a collision between m_{A} and m'_{E} under f such that m'_{E} is close to m_{E}, and it is well known that such collisions may exist for many hash functions and in fact do exist for well-known examples.[6,7] Eve can now send the message–tag pair $m'_{\mathrm{E}} + t_{\mathrm{A}}$ knowing that Bob will accept the message m'_{E} as authentic.

Searching for a collision requires Eve to have sufficient computing power, but usually, in QKD, no bounds are assumed on Eve's computing power. One should also note that the computing power needed may be lower than one would first expect.[6,7] However, even without sufficient computing power, Eve can make a list of different values of m'_{E} and the corresponding value of $z'_{\mathrm{E}} = f(m'_{\mathrm{E}}) \in \mathcal{Z}$ in advance, and save it in her device. Remember that the usual requirement of having random settings (making the message m_{E} random) does not apply to Eve; the requirement is needed to ensure that the final key is secret, something that Eve can ignore. With a prechosen m_{E}, a list of pairs $(m'_{\mathrm{E}}, z'_{\mathrm{E}})$ and her received $m_{\mathrm{A}} + t_{\mathrm{A}}$, Eve can just compute $z_{\mathrm{A}} = f(m_{\mathrm{A}})$ and pick m'_{E} from her list corresponding to z_{A}, and then send $m'_{\mathrm{E}} + t_{\mathrm{A}}$. She can even make a partial list, and simply wait for the first match to occur. In fact, the parameter ϵ_1, now interpreted as the probability that some item in Eve's list collides with m_{A}, depends linearly on the size of this list. If she is able to make a full list (one message m'_{E} for each possible z_{A}), or has sufficient computing power, she is certain of success in the sifting phase every time she performs the MITM attack.

Eve now has two sets of sifted keys, one shared with Alice and the other with Bob. The remaining steps are one-way error correction and authentication, and one-way privacy amplification and authentication. These are completed by sending random parity maps over the classical channel, and in the case of error correction also the parity values.[8–11] In the case of error correction, Eve intercepts the authenticated error-correction information (random maps and the output values) sent by Alice to Bob, and error-corrects the sifted key that she shares with Alice. She then searches for *nonrandom* maps (and the corresponding output) of the sifted key shared with Bob, which makes her message collide with Alice's under f. Note that Eve at this point may change any bit of the sifted key at the price of introducing an extra bit error into the sifted key. This will enable a collision even if all the possible maps do not. She sends the resulting message to Bob along with Alice's tag, which will then be accepted by Bob. Bob responds by an authenticated message that signals which subsets matched and which subsets were successfully error-corrected, and also indicates the error rate of the sifted key; in this simple scheme, this is used as error estimate. Eve modifies her corresponding but still waiting response to Alice so that it will collide with Bob's message under f. This may introduce some noise into the error-corrected key shared between Alice and Eve, but this goes unnoticed by Alice unless an extra detection phase is present (see below).

The privacy amplification is performed by Alice choosing a random map, and sending that over the classical channel, after which Alice and Bob apply this map to their respective reconciled keys. Here, Eve intercepts the description of the map and the tag, and privacy-amplifies the reconciled key (shared with Alice) using the received map. She then searches for a new *nonrandom* map to use for privacy amplification with Bob that makes the message coincide with Alice's under f. If Eve arranges for the reconciled key shared with Alice to be of equal length to that shared with Bob, she can even reuse the map that Alice sent. Then, Eve sends the chosen map along with Alice's tag to Bob, who will accept them and privacy-amplify his error-corrected key accordingly.

4. Countermeasures

The situation is improved if postponed authentication is used — or, for example, when using iterative reconciliation methods. More precisely, if the messages are sent in each phase as usual (sifting, error correction and privacy amplification, etc.) but not authenticated until the end of the round, then Eve's freedom to change her message is restricted to the message part in the last phase. And this severely restricts Eve's possibilities, even though an attack is still possible, as is shown in Ref. 2.

Another, more effective improvement is to use a secret key in an additional phase of the protocol. There is no explicit mention of using more secret keys for this purpose in Ref. 1 but it is implicit; it is present in the authors' Ref. 5 (here, Ref. 12). The procedure basically uses an already shared secret key to choose a hash

function to detect errors in the reconciled key. Another suggestion is to one-time-pad the reconciliation procedure.[13] Both of these suggestions are intended to keep the information leaked in error correction to a minimum, but they also implicitly add an authentication property of that phase. Using a modification like this will probably improve the situation but the needed formal proof is beyond the scope of this paper. It is perhaps important to note that this puts stronger requirements on the extra cryptographic primitives used since they are used as authentication in addition to limiting the information leakage. However, since the mentioned modifications both use cryptographically secure primitives, it is to be expected that they are resilient to extra demands of this type.

5. Conclusion

This brief review of a proposed authentication algorithm intended to rule out a man-in-the-middle attack in QKD shows that the proposed method is insecure when used in a generic QKD protocol. The main problem is that Eve is not limited to a fixed (random) message, but can in fact choose what message to send, and can check if her chosen message gives the same tag as Alice's message, since the first-step hash function f is publicly known.

Using an extra shared secret key for an extra authentication in one of the phases probably improves the situation, but it should be stressed that, unlike Wegman–Carter authentication, the security of the proposed authentication procedure is highly dependent on the context in which the authentication is applied.

Therefore, in general, great care should be taken when authentication primitives used in the context of QKD are not information-theoretically secure.

References

1. M. Peev *et al.*, A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography, *Int. J. Quant. Inf.* **3** (2005) 225 [arXiv:quant-ph/0407131].
2. T. Beth, J. Müller-Quade and R. Steinwandt, Cryptanalysis of a practical quantum key distribution with polarization-entangled photons, *Quant. Inf. Comput.* **5** (2005) 181–186 [arXiv:quant-ph/0407130].
3. M. N. Wegman and J. L. Carter, New hash functions and their use in authentication and set equality, *J. Comput. Syst. Sci.* **22** (1981) 265–279.
4. C. H. Bennett and G. Brassard, Quantum cryptography: public key distribution and coin tossing, in *Proc. IEEE Int. Conf. Comput. Syst. Signal Process.* (Bangalore, India, 1984), pp. 175–179.
5. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, UK, 2000).
6. X. Wang, Y. L. Yin and H. Yu, Finding collisions in Full SHA−1, in *Advances in Cryptology: CRYPTO 2005,* 25th Annual International Cryptology Conference (Santa Barbara, California, USA), *Lecture Notes in Computer Science*, Vol. 3621 (Springer, Germany, 2005), pp. 17–36.

7. C. D. Cannière and C. Rechberger, Finding SHA−1 characteristics: general results and applications, in *Lecture Notes in Computer Science*, Vol. 4284 (Springer, Germany, 2006), pp. 1–20.
8. C. H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, Experimental quantum cryptography, *J. Cryptol.* **5**(1) (1992) 3–28.
9. G. Brassard and L. Salvail, Secret-key reconciliation by public discussion, in *Advances in Cryptology: Eurocrypt '93*, ed. T. Helleseth, *Lecture Notes in Computer Science*, Vol. 765 (Springer, Berlin, 1994), pp. 410–423.
10. C. H. Bennett, G. Brassard and J. M. Robert, Privacy amplification by public discussion, *SIAM J. Comput.* **17**(2) (1988) 210–229.
11. C. H. Bennett, G. Brassard, C. Crépeau and U. M. Mauer, Generalized privacy amplification, *IEEE Trans. Inf. Theory* **41**(6) (1995) 1915–1923.
12. G. Gilbert and M. Hamrick, Practical quantum cryptography: a comprehensive analysis [arXiv:quant-ph/0009027].
13. N. Lütkenhaus, Estimates for practical quantum cryptography, *Phys. Rev. A* **59** (1999) 3301–3319.