Attack on a classical analogue of the Dunjko, Wallden, Kent and Andersson quantum digital signature protocol

Derrick Newton

September 15, 2015

Faculty of Engineering and Computing, Coventry University, Engineering and Computing Building, Gulson Road, Coventry, West Midlands CV1 2JH, UK

Abstract

A quantum digital signature (QDS) protocol is investigated in respect of an attacker who can impersonate other communicating principals in the style of Lowe’s attack on the Needham-Schroeder public-key authentication protocol. A man-in-the-middle attack is identified in respect of a classical variant of the protocol and it is suggested that a similar attack would be effective against the QDS protocol. The attack has been confirmed through initial protocol modelling using a automated theorem prover, ProVerif.

1 Introduction

Traditional public and private key cryptographic protocols are usually associated with the provision of confidentiality of message communication between principals (usually identified as Alice and Bob) in respect of some eavesdropping third-party (usually identified as Eve). However, the existence of confidential channels of communication between principals does not of itself necessarily imply authenticity or integrity of the transmitted message, nor does it ensure non-repudiation of the message by the sender. Signature schemes are implemented to provide authenticity, integrity and non-repudiation of messages using public key cryptography. Additionally, Lamport one-time signature schemes have also gained traction in practical situations where processor overhead is of prime concern.

Quantum digital signatures (QDS) are a current topic for research given that practical (but expensive) point-to-point quantum key distribution (QKD) is commercially available. Dunjko et al have proposed BB84-based one-time QDS schemes which do not require quantum memory. These protocols provide
a QDS scheme free of the requirement for quantum memory and processing resources, contrasting earlier schemes of Gottesman and Chuang. In this paper we investigate the QDS protocol P2 of Dunjko et al by consideration of a classical analogy of the quantum protocol. We identify a possible man-in-the-middle attack on the classical digital signature protocol inspired by Gavin Lowe’s attack on the Needham-Schroeder protocol and we propose a security attack on the QDS protocol by analogy. Further, we are currently modelling the classical P2 protocol in the Applied Pi Calculus and analysing the protocol using an automated theorem prover called ProVerif. We discuss some initial findings from reachability experiments in support of our attacker model.

2 Protocol P2

Dunjko et al’s Protocol P2 is simplified and stated in classical form as follows:

Key distribution stage

1. For each possible future message \( m_f = 0, 1 \), Alice (A) generates two different secret keys consisting of sequences of classical bits.

2. For each possible message \( m_f = 0, 1 \), Alice sends one secret key to Bob (B) and the other to Charlie (C) via secure classical channels.

3. For each signature element and for \( m_f = 0, 1 \), Bob (Charlie) randomly chooses to either keep it or send it to Charlie (Bob) via a secure classical channel. Essentially, Bob (Charlie) applies a bitstring mask operation against the key bitstring: \( \text{mask} (k_{m_f B}, n_{m_f B}) \) (resp. \( \text{mask} (k_{m_f C}, n_{m_f C}) \)), where \( n_{m_f B} \) (resp. \( n_{m_f C} \)) is Bob’s (Charlie’s) chosen bitstring mask for future message bit \( m_f \).

Messaging stage

1. To send a signed one-bit message \( m \), Alice sends \( (m, k_{mB}, k_{mC}) \) to Bob, say, where \( k_{mB}, k_{mC} \) are the secret keys assigned to Bob (resp. Charlie) corresponding to the message \( m \).

2. Bob checks \( (m, k_{mB}, k_{mC}) \) against his key and the partial key sent to him by Charlie and accepts if matched.

3. Bob now forwards message \( (m, k_{mB}, k_{mC}) \) to Charlie and Charlie checks this against his key and the partial key sent by Bob and accepts if matched.

Communication between the three principals over secure channels within the protocol is represented by table II.
1. \( A \rightarrow C : k_{0C}, k_{1C} \)
2. \( A \rightarrow B : k_{0B}, k_{1B} \)
3. \( B \rightarrow C : \text{mask}(k_{0B}, n_{0B}), \text{mask}(k_{1B}, n_{1B}) \)
4. \( C \rightarrow B : \text{mask}(k_{0C}, n_{0C}), \text{mask}(k_{1C}, n_{1C}) \)
5. \( A \rightarrow B : m, k_{mB}, k_{mC} \)
6. \( B \rightarrow C : m, k_{mB}, k_{mC} \)

Table 1: Classical P2 protocol

3 An attack on the P2 protocol

In order to model the principle components of the P2 protocol we use the classical statement of the protocol where point-to-point quantum channels are replaced by secure classical channels.

3.1 Possible attack by \( E \) against \( B \)

We will now consider a possible man-in-the-middle attack by Eve on this protocol. Eve (\( E \)) is a member of the communicating network. She is trusted sufficiently by the other principals so that secure communication channels can be established between her and her fellow principals. Eve takes control over Bob’s incoming and outgoing communications so that Alice and Charlie send messages to Eve in the belief that they are talking to Bob. Bob, on the other hand, receives messages from Eve believing these messages to be originating from Alice or Charlie. In this position Eve can choose to flip signatures and message bits so that Bob receives a different message from Alice than was originally sent and yet Bob is able to verify the message against its attached signature. From Charlie’s perspective, however, the message received agrees with that which was sent by Alice. The details of this attack are set out below and the communications are presented in table 2.

Key distribution stage

1. For each possible future message \( m_f = 0, 1 \), Alice (\( A \)) generates two different secret keys consisting of sequences of classical bits.
2. For each possible message \( m_f = 0, 1 \), Alice sends one secret key to Eve (\( E \)) pretending to be Bob (\( B \)) and the other to Charlie (\( C \)) via secure classical channels.
3. For each possible message \( m_f = 0, 1 \), Eve (\( E \)) swaps the secret keys so that the key \( k_{0B} \) is assigned to future message bit \( m_f = 1 \) and key \( k_{1B} \) is assigned to future message bit \( m_f = 0 \) (now identified as \( k'_{0B}, k'_{1B} \)).
4. Bob applies random bit selection to his keys, i.e. he assigns bitstring masks to each key for future message \( m_f \) and sends the partial keys \( k_{part0B}, k_{part1B} \) to Eve thinking that he is communicating with Charlie.

5. Eve has knowledge of the complete keys for Bob and the partial keys so she is able to compute the effect of the bitstring masks on the original keys. She sends these “restored” partial keys \( k_{part0B}, k_{part1B} \) to Charlie who believes that he is receiving them from Bob.

6. Charlie sends his masked partial keys \( k_{part0C}, k_{part1C} \) to Eve in the belief that he is sending them to Bob.

7. Eve swaps the partial keys sent by Charlie so that the partial key assigned to future message \( m_f = 0 \) is now assigned to \( m_f = 1 \) and vice versa. The swapped partial keys \( k_{part0C}', k_{part1C}' \) are then sent to Bob.

**Messaging stage**

1. Alice sends her one bit signed message \((m, k_{mB}, k_{mC})\) to Eve thinking that she is communicating with Bob.

2. Eve flips the message: \( m = 0 \) is replaced by \( m' = 1 \) (or vice versa).

3. Bob matches the signature to the swapped key \( k_{mB}' \) and to the swapped partial key from Charlie \( k_{partmC}' \) and accepts if matched.

4. Bob sends the flipped message and signatures \((m', k_{mB}, k_{mC})\) to Eve thinking that he is communicating with Charlie.

5. Eve flips the message again so that \( m' = 1 \) is replaced by \( m = 0 \) (or vice versa).

6. Charlie confirms the (original) message signature against his own signature and against the partial signature received from Eve (assumed to be Bob) which is the corrected partial signature consistent with Alice’s original signature.

**Result** Following this attack Charlie has received the correct message \( m \) from Alice and has assured himself of its authenticity by verification of the signature. Bob has received the flipped message \( m' \) from Alice and assured himself of its authenticity by verification of the signature. Consequently, authenticity and integrity of the message have not been provided by the signature protocol. A similar attack can be devised against Charlie.
1. $A \rightarrow C : k_{0C}, k_{1C}$
2. $A \rightarrow E(B) : k_{0B}, k_{1B}$
3. $E(A) \rightarrow B : k'_{0B}, k'_{1B}$ swap $k_{0B}, k_{1B}$
4. $B \rightarrow E(C) : k_{0B}, k_{1B}$ swap $k_{1B}, k_{1B}$

where $k_{0B} = \text{mask}(k'_{0B}, n_{0B}), k_{1B} = \text{mask}(k'_{1B}, n_{1B})$

5. $E(B) \rightarrow C : k_{0B}, k_{1B}$

where $k_{0B} = \text{mask}(k'_{0B}, n_{0B}), k_{1B} = \text{mask}(k'_{1B}, n_{1B})$

6. $C \rightarrow E(B) : k_{0C}, k_{1C}$

where $k_{0C} = \text{mask}(k'_{0C}, n_{0C}), k_{1C} = \text{mask}(k'_{1C}, n_{1C})$

7. $E(C) \rightarrow B : k_{0C}, k_{1C}$ swap partial keys

where $k_{0C} = \text{mask}(k'_{0C}, n_{0C}), k_{1C} = \text{mask}(k'_{1C}, n_{1C})$

8. $A \rightarrow E(B) : m, k_{mB}, k_{mC}$
9. $E(A) \rightarrow B : m', k_{mB}, k_{mC}$ swap $m, not(m)$
10. $B \rightarrow E(C) : m', k_{mB}, k_{mC}$
11. $E(B) \rightarrow C : m, k_{mB}, k_{mC}$ swap $m, not(m)$

Table 2: Man-in-the-middle attack against $B$ in the P2 protocol

4 Formal modelling of the protocol and the attack

Research into formal modelling of the protocol is ongoing. However, the findings of this paper are supported by reachability experiments performed over the classical P2 protocol.

The classical P2 protocol is encoded in the Applied Pi Calculus of Abadi and Fournet [1], as modified by Blanchet, Smyth and others and implemented in ProVerif [2]. ProVerif is a powerful automated protocol verification tool for reachability, secrecy, correspondence properties and observational equivalence. Experiments are performed with the protocol model to validate the attack outlined above. Full details of the formal modelling analysis will follow in a subsequent paper. However, initial experiments support the existence of the attack as outlined above.

5 Application to QDS

The attack outlined above applies to the classical cryptographic interpretation of protocol P2. However, it can be observed that the attack involves reassignment of keys to future messages and application of Bob’s key mask so we suggest that the attack could be transferred to the QDS protocol P2 in which key distribution is established using quantum channels. No direct observation and subsequent collapse of the key distribution qubits by Eve is required. It remains an open
research question as to whether or not compromise to quantum P2 or indeed to protocol P1 can be verified by formal modelling.

In this research we have not, as yet, considered modelling using process algebras developed for quantum communications. We are mindful of the research which has been carried out in respect of quantum process algebras by Gay and Nagarajan with CQP [5], Feng et al with qCCS [9] but the translation of correspondence and reachability assertions to quantum protocol modelling has not, to our knowledge been established to date. Indeed there are fundamental questions as to the application of event labelling and process dependencies to quantum protocol models which require resolution before we can devise correspondence and reachability tests within quantum models.

6 Conclusions

In this paper we have presented an attack on the classical implementation of Dunjko et al’s signature protocol P2. The attack allows an eavesdropper to modify a signed message and and swap signatures within the protocol so that a principal is able to authenticate the message and signature even though the message has been altered. Additionally, we suggest that this attack would extend to a quantum digital signature version of the P2 protocol as the eavesdropper is not required to observe the key messages on the quantum channel.

A detailed analysis of the classical protocol using an automated theorem prover, ProVerif is ongoing. Extending the work further to formal modelling and analysis using quantum process algebras or modifications to classical process algebras to encapsulate quantum processes is a further goal of this research.

References

[1] Martin Abadi and Cedric Fournet. Mobile values, new names, and secure communication. SIGPLAN Not., 36(3):104–115, 2001.

[2] Bruno Blanchet, Ben Smyth, and Vincent Cheval. ProVerif 1.90: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial, May 2015.

[3] Vedran Dunjko, Petros Wallden, and Erika Andersson. Quantum digital signatures without quantum memory. Physical Review Letters, 112, 2014.

[4] Vedran Dunjko, Petros Wallden, Adrian Kent, and Erika Andersson. Quantum digital signatures with quantum key distribution components. arXiv, (1403.5551v2), November 2014.

[5] Simon J. Gay and Rajagopal Nagarajan. Communicating quantum processes. In Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages, pages 145–157. ACM Press, New York, 2005.
[6] Daniel Gottesman and Isaac Chuang. Quantum digital signatures. arXiv:quant-ph/0105032, 2001.

[7] Leslie Lamport. Constructing digital signatures from a one way function. Technical report, Computer Science Laboratory, SRI International, October 1979.

[8] Gavin Lowe. An attack on the needham-schroeder public-key authentication protocol. Information Processing Letters, 56(3):131–133, 1995.

[9] Mingsheng Ying, Yuan Feng, Runyao Duan, and Zhengfeng Ji. An algebra of quantum processes. ACM Transactions on Computational Logic, 10(3):19:1–19:36, 2009.