Model Checking Quantum Key Distribution Protocols

Baichuan Huang[1], Yan Huang[1], Jiaming Kong[1], Xin Huang[2]

[1] Department of Computer Science, University of Liverpool
{B.Huang6, Y.Huang58, J.Kong4} @student.liverpool.ac.uk
[2] Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University
Xin.Huang@xjtlu.edu.cn

***Abstract*— Quantum key distribution protocols use quantum information theories to guarantee the security of key exchange procedure, and model checking is a verification technique which could be used to test the security of it. In this paper, a new group quantum key distribution protocol is designed based on BB84 protocol, which is a possible solution to handle the security issue in communication between multi-users. Discrete time Markov chain and probabilistic computation tree logic are used to model the protocol procedure and verify its security properties in PRISM. Also, we provide the theoretical proof for this group protocol, which supports the model checking results from PRISM. Our study gives an insight into some of the major difficulties of quantum security protocol design and analysis.**

Keywords-component; BB84; model checking; group QKD

I. INTRODUCTION

In cryptography, the group key shared among multi-users, which allows a bunch of users communicate with each other safely, and guarantees other people not in the group cannot decrypt those messages [11]. Because the group key has to be shared among all group members, a secure method to share this group key is necessary, and the photon may be the media to do so.

Quantum physics ensures the security of Quantum Key Distribution (QKD), and the BB84 protocol is the first QKD protocol proposed in 1984 [2]. There are two papers which carefully explored how the protocol works, how this protocol might be attacked by eavesdroppers, as well as provided proof of the security of BB84 [4][6]. Also, there is another way to verify the security of QKD protocols, which is the model checking [3].

Model checking is a formal verification technique to test all possible states of system model in a brute-force way and validate whether a presented model satisfies specifications [5]. PRISM is an effect probabilistic model checker to verify the quantum information transmission system which also appears the feature of probabilities, so it can be used to model and check the probabilistic behavior of systems. PRISM can also be used to compute the probability of a formula Φ satisfied by a model, for example, PRISM can compute the value of

$$P_{\sigma,\Phi} = \Pr\{\sigma \models \Phi\}$$

for the given σ and Φ [1]. Therefore, PRISM could be the tool to demonstrate and research the implementation of QKD in group key.

QKD is a possible solution to share the group key among group members. Now, only single information transmission channel which applies BB84 protocol has been studied widely, the research about multiple channels which applies BB84 protocol as a group is inadequate. Once this new QKD group is viable, it may bring abundant use value in safety communication area. Hence, we firstly model checks the BB84 protocol on PRISM, then we design a group QKD protocol called tree model which extends BB84 to distributed group key. Moreover, we demonstrate how to validate the security of our proposed group QKD protocol in model checking. As part of it, theory proof is displayed as well.

This paper is organized as follows. In section II, we present the overview of BB84 and how it is model checked as prerequisites for the group QKD protocol and its model checking procedure. Section III shows how the group QKD tree model protocol is modeled in PRISM. We also analyze the security of tree model based on model checking results and validate that these results are theoretically correct in section IV. We conclude our work in the final section.

II. BB84 AND MODEL CHECKING

A. BB84 Overview

In the scheme of BB84, the sender Alice transmits quantum bits in the rectilinear $\oplus$ or diagonal $\otimes$ basis to the receiver Bob, who measures these bits using the same basis [7]. A quantum bit can be represented in one of the quantum states below:

$$|\psi(0,\oplus)\rangle = |0\rangle$$
$$|\psi(1,\oplus)\rangle = |1\rangle$$
$$|\psi(0,\otimes)\rangle = |+\rangle$$
$$|\psi(1,\otimes)\rangle = |-\rangle$$

The process of BB84 protocol for exchanging the common key has two parts [10]. The first part includes three steps:

- Firstly, Alice uses either $\oplus$ or $\otimes$ polarization to prepare photons randomly, so that she can transmit

978-1-5090-3906-7/16 $31.00 © 2016 IEEE
DOI 10.1109/ITME.2016.155

photons in four polarization states, which are 0, 45, 90 and 135 degrees.

- Secondly, after recording the polarization of these photons, Alice sends them to Bob.
- Thirdly, Bob records both the polarization of photons he has received by using the $\oplus$ or $\otimes$ basis randomly. Since he does not know whether the photon is measured on the same basis as what Alice used, he has a 50% probability to be lucky to choose the same quantum bit, in which case the quantum bit should agree with that sent by Alice; or, he will choose the basis which is different from what Alice chose and might acquire the same bit with a 50% probability. After that, Bob gets a sequence of binary bits.

In the second part, Alice and Bob continue to communicate through the public channel:

- First, Alice chooses parts of photons as the check part (the other part as the key part) and tells her choice to Bob.
- Secondly, Alice and Bob reveal both bases and bits of photons of the check part, if the number of photons for which Alice and Bob choose the same basis, but they obtain different bits exceeds a bound (because of channel noise, there are some photons that could have different bits even they are measured on the same basis), then this dialog should be aborted or restart.
- Thirdly, they reveal the bases of photons from the key part and choose the sequence of a bit of photon that has the same basis as a common key.

Once the bound is exceeded in the check part, indicating that there is interference on this channel, and probably an eavesdropper exists between Alice and Bob, this communication is no longer secure [10].

We consider one possible attack: intercept-resend attack [8]. The eavesdroppers intercept and capture the photon with the state $|\rho(\alpha_n,\beta_n)\rangle$ from the channel sent by Alice, the subscript *n* is the index of photons in this transmission. Then, the eavesdropper chooses a basis $\bar{\beta}_n$ to measure the photon to obtain the bit of photon $\bar{\alpha}_n$, if $\bar{\beta}_n = \beta_n$, then $\bar{\alpha}_n = \alpha_n$. Otherwise, due to the quantum mechanism (Photon polarization), $\bar{\alpha}_n$ is equal to α_n with half probability. Next, the photon with the state $|\rho(\bar{\alpha}_n,\bar{\beta}_n)\rangle$ is put back to the channel and transformed to Bob. Here we denote $|\rho(\alpha'_n,\beta'_n)\rangle$ as the state of the photon that Bob uses basis β'_n to measure and obtain bit α'_n from the channel [9].

B. Organize BB84 in Model Checking

Discrete-time Markov chains (DTMCs) is chosen as the probabilistic model type in PRISM. Its process is from one state to another, where the transitions are not sequential [12]. Two modules Alice and Bob respectively represent as the sender and receiver in QKD. A Channel module is created to represent as the photon transmission channel. Alice firstly puts the photons into the Channel, and Bob receives the photons from the Channel. This process is presented in PRISM as follows [5]:

$$[\text{action}]\text{condition} \rightarrow a_1 : (\text{var}_1' = \text{value}_1) + a_2 : (\text{var}_2' = \text{value}_2) + \ldots$$
$$\ldots + a_n : (\text{var}_n' = \text{value}_n);$$

Inside the square brackets, it is the label name of this action. We can write different actions in one module, but with the same label name. Moreover, this label name can force two or more modules to make transitions simultaneously. The condition next to the label name is the guard, which decides whether this action can or cannot proceed. Once the condition is satisfied, the procedure after right arrow will start. In this part, var_i is the local variable, which is updated to the value_i with the probability a_i, and we should ensure $\sum_{i=1}^{n} a_i = 1$.

Moreover, there is an Eve module served as eavesdropper to attack the channel. To be clear, the channel noise is not considered in this paper.

To complete the model checking, one property formula

$$\partial = (\beta_n = \beta'_n) \wedge (\alpha_n \neq \alpha'_n)$$

is used to compute the probability that Eve is detected when N photons are transmitted. It stands for that Alice and Bob use the same basis while they have different states. The results from this formula can prove the security of BB84 protocol and further for the group QKD. Corresponding to PRISM, the code can be written in the form of

$$P = ?[\Phi_1 \bigcup \Phi_2]$$

This "Until" path property means “unbounded until” and the expression $\Phi_1 \bigcup \Phi_2$ means that Φ_1 is continuously holding in the current state until Φ_2 becomes true. Hence, ∂ will be transformed in Φ_2, and Φ_1 will be the state that Φ_2 does not hold at the moment.

III. MODEL CHECKING GROUP QKD

In this section, we will present the framework of group QKD: tree model, and the implementation of this model in PRISM.

A. Group QKD: Tree Model

The tree model is presented as Figure 1, where Alice or the User 1 is the start of this protocol. Photons are created by them and then transmitted to following layers, so all of them can share the same key and communicate freely and securely. We will mainly consider the three parties as a group (the three-

party structure is the fundamental unit in tree model), in which case Alice simultaneously distributes her two same photons to Bob and Clair, while there is no communication between the receivers. During the transmission, both channels will be attacked by Eve. The concrete steps of tree model are as below:

- Alice selects a random string of bits $\alpha \in \{0,1\}^N$ and a random string of bases $\beta \in \{\oplus,\otimes\}^N$ as shown in table 1, where N is the length of the key part.
- Alice prepares two same quantum photons which are in the quantum state $|\rho(\alpha_n,\beta_n)\rangle$ each time and sends one to Bob and one to Clair through their quantum channels.
- After Bob and Clair receive the qubits, they separately measure the qubits in either $\oplus$ or $\otimes$.
- Finally, Alice publishes the bases she chose and verifies bits with Bob and Claire.

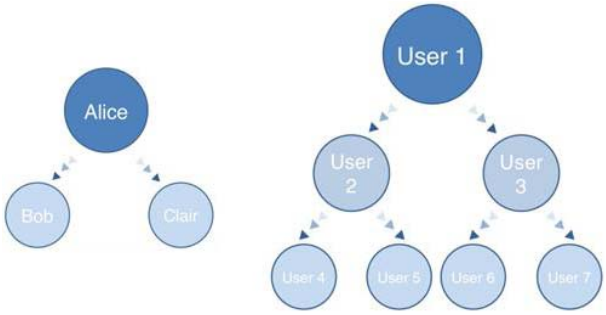

Figure 1. An example of the tree model.

TABLE I. THE INFORMATION BIT AND BASIS

Bit	*Basis*
0	$\oplus$
1	$\otimes$

B. PRISM Implementation

To construct the tree model in PRISM, we use DTMCs as the model type and create seven modules, which are Alice, Bob, Claire, Channel 1, Channel 2, Eve 1 and Eve 2. We use variables *bas* and *bit* to store information of the photon, and a *state* variable to indicate which process of transmission the module is in. Moreover, compared with other modules, the module Alice has one more variable *alice_index*, which means the number of photons already have been transmitted.

Because we separate seven modules to simulate seven distinct characters in a real quantum transmission system, the synchronization is critical to associate different module transitions simultaneously. For example, each time when a photon is transmitted, the senders, receivers and eves will firstly choose which basis to perform the measurement, and Alice has one more step to do: choose the bit of the photon. To ensure those initializations are simultaneously completed, and no other actions would be performed before them, a label *[startup]* is used. All actions labeled with same label name will run concurrently based on the mechanism of PRISM.

Next, Alice puts photon into the channel, so we write the first action into Alice module, and the second one into Channel module. After that, the variables *alice_state* and *ch_state* are changed and they indicate that this model could enter the next state of transmission. As following code:

```
Alice : [aliceput](alice_state=1) -> (alice_state'=2);

Channel1 : [aliceput](ch1_state=0) -> (ch1_state'=1) &
(ch1_bas'=alice_bas) & (ch1_bit'=alice_bit);

Channel2 : [aliceput](ch2_state=0) -> (ch2_state'=1) &
(ch2_bas'=alice_bas) & (ch2_bit'=alice_bit);
```

When the photon is in the channel, Eve starts to measure the photon. Here, we will compare β_n and $\bar{\beta}_n$ (i.e. *ch_bas* and *eve_bas* in PRISM). If $\beta_n == \bar{\beta}_n$ (i.e. *ch_bas == eve_bas* in PRISM), then the photon will retain its information. If not, the operation should be like this (the LUCKY is 0.5, which is the probability of obtaining correct bit by using the wrong basis to measure the photon):

```
Channel : [eveget](ch_state=1) -> (ch_state'=2);

Eve : [eveget](eve_state=1) & (eve_bas!=ch_bas) ->
LUCKY : (eve_state'=2) & (eve_bit'=ch_bit)+
(1-LUCKY) : (eve_state'=2) & (eve_bit'=1-ch_bit);
```

After that, Eve puts the photon with $|\rho(\bar{\alpha}_n,\bar{\beta}_n)\rangle$ back to the channel. Bob and Claire measure the photon in the same manner as Eve has done.

Finally, Alice and Bob reveal their choices of basis and use the information reconciliation technique to validate the bit of photon. If the formula ∂ does not hold between Alice and Bob or Alice and Claire, which means no Eve is detected, then Alice sends next photon until all photons have been transmitted safely.

When Alice has no photons, the probability of "∂ hold" is calculated. We transform this mathematical form into PRISM language as: P=?[true $\cup$ alice_state=5] .

IV. RESULTS

Model checking has been completed, and the results will be discussed. This paper also verifies the results produced by PRISM theoretically.

A. Model Checking Results

The security of quantum key distribution for multiple users based on BB84 protocol is reliable. We analyze the tree model protocol according to the PRISM graphs. In this model checking, we choose N= [4, 28] and LUCKY=0.5 as the constants for the "experiment" to generate a probability graph.

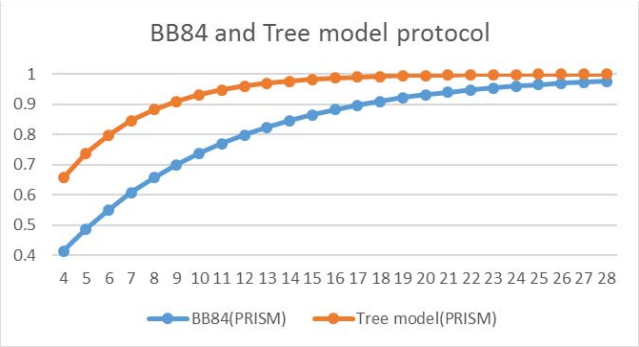

Figure 2. The PRISM result for the BB84 and tree models.

TABLE II. THE PROBABILITY THAT DETECTS OF EVE

Model/Number of photons	*N=4*	*N=10*	*N=16*	*N=22*	*N=28*
BB84	0.414	0.737	0.882	0.947	0.976
Tree model	0.656	0.931	0.986	0.997	0.999

The results of BB84 are identical to results in tree model, and the results of the tree model are within expectation as well. Next, we will mainly analyze the security of this group protocol.

As Figure 2 shows, the process of generating a shared key is secure in the tree model group protocol. The reason why the probability of detecting eve is higher than the normal BB84 is that the property we used in formula (1) indicates that this property is always true until Eve is detected in either one or both channels. When N=28, the probability is 0.999 (near to 1, shown in table 2), which means we probably still cannot detect the existence of Eve.

Further, if the key part has the same length as the key generated by BB84, more photons need to be transmitted because, in this tree model, there are three parties which require owning the same information of photons rather than two parties. In another case, if only one channel is attacked during the transmission, the probability of detecting Eve will be the same as the normal BB84 protocol. Finally, as a group protocol of generating the shared key, the tree model is at least as secure as BB84.

B. Theoretical Verification

In this part, we will prove that the results from PRISM are theoretically correct. We start with the basic BB84 protocol and assume that Eve uses the intercept-resend attack.

1) BB84: As a precondition, the probability that Alice and Bob choose the same basis is 0.5. Here we use BS to denote this probability that two parties choose the same basis for generic proof, and NBS as the opposite condition that $\mathrm{NBS} = 1 - \mathrm{BS}$. We do not consider the case that they choose different basis because Eve cannot be detected in such condition. Moreover, the probability that Eve "luckily" chose the same basis as Alice and Bob is also BS (0.5), in which situation Eve will not disturb the state of the photon. When Eve is "unlucky", the basis he chooses will be different from Alice's and Bob's. Certainly, in this case, the state of the photon will be changed, as well as the bit in the photon. Thus, there is the half probability that the basis Bob uses is different from what the photon has now. Therefore, when Alice and Bob choose the same basis, the probability that Eve is detected is $(1-\mathrm{BS})/2$ (Eve unluckily chooses the different basis, and Bob obtains wrong bit). In addition, we use P to denote the probability that Eve does not be detected, and NP to denote the probability that Eve is detected. Thus, we have:

$$\mathrm{P} = \frac{1+\mathrm{BS}}{2}$$

$$\mathrm{NP} = \frac{1-\mathrm{BS}}{2}$$

In case one, we consider only one photon is sent from Alice to Bob, and Eve is detected. This means Alice and Bob choose the same basis, while Eve chooses the different basis. Thus, Bob obtained different bit compared to Alice's bit from the changed photon after Eve disturbs it. The probability of this case is $\mathrm{BS} \times \mathrm{NP}$.

In case two, Alice sends two photons to Bob, and Eve is detected when the second photon is transferred. Therefore, Eve is not detected for the first photon, which means Alice and Bob choose different photons, or Eve is "lucky". The probability of such case to happen is $\mathrm{NBS} + \mathrm{BS} \times \mathrm{P}$. Next, for the second photon, the situation that Eve is detected is the same as case one. In total, the probability of case two to happen is $(\mathrm{NBS} + \mathrm{BS} \times \mathrm{P}) \times (\mathrm{BS} \times \mathrm{NP})$.

As an induction, we can conclude that the probability of Eve to be detected when the n_{th} photon is sent from Alice to Bob (suppose that the n-1 photons do not detect any eavesdropper) is $(\mathrm{NBS} + \mathrm{BS} \times \mathrm{P})^{n-1} \times (\mathrm{BS} \times \mathrm{NP})$.

Finally, after we synthesize all above conditions, we can have a probability formula of Eve being detected when Alice sends n photons to Bob:

$$
\begin{aligned}
&\mathrm{BS}\times\mathrm{NP}+(\mathrm{NBS}+\mathrm{BS}\times\mathrm{P})\times(\mathrm{BS}\times\mathrm{NP})+\\
&(\mathrm{NBS}+\mathrm{BS}\times\mathrm{P})^2\times(\mathrm{BS}\times\mathrm{NP})+\ldots\\
&+(\mathrm{NBS}+\mathrm{BS}\times\mathrm{P})^{n-1}\times(\mathrm{BS}\times\mathrm{NP})\\
&=1-(\mathrm{NBS}+\mathrm{BS}\times\mathrm{P})^n\\
&=1-(1+\frac{\mathrm{BS}^2}{2}-\frac{\mathrm{BS}}{2})^n\\
&=\mathrm{PB}
\end{aligned}
$$

and we use *PB* to denote this formula.

2) Tree Model: The tree model of BB84 protocol basically combines two original BB84 protocols. One is Alice to Bob, and the other one is Alice to Claire. Eve could be detected in any one of these channels means that eavesdroppers are detected in this tree model QKD. Therefore, to simplify, the probability of Eve being detected in this model can be computed by its adverse condition that the probability of Eve not being detected. To verify the results of tree model from PRISM, the formula we have derived in BB84 can be applied. Therefore, the probability of Eve being detected in tree model is:

$$1-(1-\mathrm{PB})^2 .$$

With the $\mathrm{BS}=0.5$, we obtain:

$$1-(\frac{49}{64})^n$$

.

We have validated this with the PRISM results, and they can match at least nine decimals. As a conclusion for those double validations, when the number of photons is more than 28, Eve will be detected with extremely high probability, and the key produced in this transmission will be disregarded. Thus, we can confirm that the tree model is secure in group QKD.

V. CONCLUSION

This paper demonstrated how to validate the security of QKD protocol. Then, it proposed a group QKD protocol, which is a tree model protocol based on the well-known quantum cryptography BB84. We applied model checking as a verification technique to prove that this group protocol is at least as secure as BB84 with only one eavesdropper.

ACKNOWLEDGMENT

This work has been supported by the XJTLU research development fund projects RDF140243 and RDF150246, the Suzhou Science and Technology Development Plan under grant SYG201516, and Jiangsu Province National Science Foundation under grant BK20150376. Also, we appreciate the help of Jie Zhang to revise this paper.

REFERENCES

[1] Gay, S., Nagarajan, R., & Papanikolaou, N. (2005). Probabilistic Model Checking of Quantum Protocols. arXiv preprint quant-ph/0504007.

[2] Bennett, C. H. (1984). Quantum cryptography: Public key distribution and coin tossing. In International Conference on Computer System and Signal Processing, IEEE, 1984 (pp. 175-179).

[3] Elboukhari, M., Azizi, M., & Azizi, A. (2010). Verification of Quantum Cryptography Protocols by Model Checking. Int. J. Network Security & Appl,2(4), 43-53.

[4] Mayers, D. (2001). Unconditional security in quantum cryptography. Journal of the ACM (JACM), 48(3), 351-406.

[5] Elboukhari, M., Azizi, M., & Azizi, A. (2010). Analysis of quantum cryptography protocols by model checking. Int. J. Universal Comput. Sci, 1, 34-40.

[6] Shor, P. W., & Preskill, J. (2000). Simple proof of security of the BB84 quantum key distribution protocol. Physical review letters, 85(2), 441.

[7] Bennett, C. H., Bessette, F., Brassard, G., Salvail, L., & Smolin, J. (1992). Experimental quantum cryptography. Journal of cryptology, 5(1), 3-28.

[8] Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., Dušek, M., Lütkenhaus, N., & Peev, M. (2009). The security of practical quantum key distribution. Reviews of modern physics, 81(3), 1301.

[9] Papanikolaou, N. K. (2004). Techniques for design and validation of quantum protocols (Doctoral dissertation, University of Warwick. Department of Computer Science).

[10] Lo, H. K., & Chau, H. F. (1999). Unconditional security of quantum key distribution over arbitrarily long distances. science, 283(5410), 2050-2056.

[11] Katz, J., & Yung, M. (2003, August). Scalable protocols for authenticated group key exchange. In Annual International Cryptology Conference (pp. 110-125). Springer Berlin Heidelberg.

[12] Alur, R., & Henzinger, T. A. (1990, June). Real-time logics: Complexity and expressiveness. In Logic in Computer Science, 1990. LICS'90, Proceedings., Fifth Annual IEEE Symposium on e (pp. 390-401). IEEE.