Position-Verification in Multi-Channel Model

Huajun Zhang, Zongyang Zhang and Zhenfu Cao

Shanghai Jiao Tong University, Email: zhanghuajun.cn@gmail.com

Abstract. We propose an efficient collusion-attack-resistant position-verification protocol in a new model named multi-channel model. In the multi-channel model, there are lots of communication channels. When a player picks a random channel and sends a short message over it, the message might slip by an adversary with high probability if the adversary does not know the channel beforehand. This idea is motivated by the multiple-access methods. We adopt it to solve the position-verification task. Adding different constraints into the multi-channel model, we make three sub-models: receiving-constrained multi-channel model, sending-constrained multi-channel model and both-constrained multi-channel model. Our position-verification protocol is secure under all of these sub-models with appropriate parameters.

Keywords: Multiple-access method, Position-verification.

1 Introduction

1.1 Position-Verification

Recently, some researchers [SSW03] try to find a solution to the question that *how to prove where you are*, i.e., position-verification. Position-verification protocols enable a prover to communicate with a group of verifiers and give them an interactive proof of its geographic position. Position-verification is the foundation of position-based cryptography [CGMO09]. It is also a hot topic in the field of wireless security [SSW03,CH05,SP05,VN06,CCS06,ZLFW06,CHH09]. Most previous protocols are insecure against *collusion attacks*, i.e., a set of adversaries can collude together to prove that someone of them is at position P, but in fact, no one is there.

In [CHH09] and [CGMO09], from different points of view, they prove that it is impossible to propose a collusion-attack-resistant protocol of position-verification in the typical model without additional assumptions. This impossibility result holds even if the adversaries are assumed to have limited computation power [CGMO09]. And, it also holds in the quantum setting [BCF$^+$10]. Thus, to propose a secure position-verification protocol, we have to find some appropriate assumptions, first.

Chandran, Goyal, Moriarty and Ostrovsky [CGMO09] propose a position-verification protocol in the bounded retrieve model (BRM) which can resist collusion attacks. It is an information-theoretically secure protocol. However, it is not efficient enough. In that protocol, the verifiers need to broadcast large bursts of information that might be hard to do. In addition to the position-verification, they also propose several position-based key-exchange protocols. After the execution of these protocols, a shared key among the verifiers and the prover is established. Based on these key exchange protocols, lots of position-based tasks can be fulfilled.

Next to that, Buhrman et al. [BCF$^+$10] propose another position-verification protocol in the quantum setting. The security of this protocol is based on the assumption that the adversaries are not allowed to have pre-shared entangled quantum states (No-PE model). In the same paper, they bridge the position-verification and the position-based key-exchange by a new primitive: position-based authentication. Position-based authentication protocols enable the verifiers to make sure that

a given message originates from the prover at the claimed position. They show a general method to propose a position-based authentication protocol based on a position-verification protocol. A position-based key-exchange protocol can be proposed by combining a position-based authentication protocol and a general authenticated key-exchange protocol. Thus, position-verification becomes the foundation of all these things. If we find a good position-verification protocol, we can solve these position-based tasks one by one.

1.2 Multiple-Access and Multi-Channel

Our approach is motivated by the multiple-access methods (or channel access methods). In telecommunication and computer networks, a multiple-access method enable several terminals connected to the same transmission medium to transmit over it and to share its capacity. Variety of multiple-access methods are used in wireless communication area such as frequency division multiple access (FDMA), time division multiple access (TDMA), code division multiple access (CDMA). In these multiple-access methods, the wireless communication environment is divided into lots of channels by different means. In FDMA, each channel has a typical frequency which is different from others. In TDMA, different channels are composed by different time-slots. In CDMA, a channel is defined by a particular random code.

The methods to send/receive messages in different channels are different in these multiple-access methods. That means sending/receiving in two different channels at the same time costs twice as much as that in one channel. Thus, a user's communication ability is possibly constrained. We usually consider all the channels in a system as a big channel if the players have unlimited communication power. An unlimited player can access to all the channels at the same time. The more channels there are, the better communication capability it has. However, more channels are not good for a player with limited communication power. It means more information out of its control if the total amount of the channels is larger than the amount that the player can access. Based on this observation, we build the MCM.

1.3 Our Results

In this paper, we give the following results.

Multi-channel Model for Position-verification. We try to build an appropriate model called multi-channel model (MCM) for the position-verification task in multi-channel scenarios. In the MCM, we define a sending oracle and a receiving oracle to handle all the communication jobs so that the complex discussions of communication details are avoided. To characterize the communication abilities of the players, we define two kinds of ideal communication devices: atomic sending devices and atomic receiving devices. By constraining the maximal number of the atomic devices that each party controls, we give three sub-models: sending-constrained multi-channel model (sc-MCM), receiving-constrained multi-channel model (rc-MCM) and both-constrained multi-channel model (bc-MCM).

Position-verification Protocol. We propose a new position-verification protocol in the MCM. We use directional messages in this protocol. In the wireless scenarios, using directional messages can decrease the interference chance and save the output power of the sender. We prove that our protocol is secure against collusion-attacks in all the sub-models of the MCM with appropriate parameters. And the efficiency of our protocol is also acceptable.

1.4 Paper Structure

In Section 2, we define the MCM, the sub-models of the MCM and the position-verification task. In Section 3, we show our position-verification protocol and prove its security under the rc-MCM, sc-MCM and bc-MCM. In Section 4, we compare the MCM with some leakage models. Then, we show some tradeoff tricks of our protocol to improve its efficiency or security.

2 The Model

There are many wireless communication channels in the real world. The methods to send/receive messages in different channels are different. That means sending/receiving in two different channels at the same time costs twice as much as that in one channel. Thus, a user's communication ability is possibly constrained.

We usually consider all the channels in a system as a big channel if the players have unlimited communication power. An unlimited player can access to all the channels at the same time. The more channels there are, the better communication capability it has. However, more channels are not good for a player with limited communication power. It means more information out of its control if the total amount of the channels is larger than the amount that the player can access. Based on this observation, we build the MCM.

2.1 Multi-Channel Model

All the messages in the MCM are carried by the radio waves. We assume that the radio waves transmit in space at the speed of light c. For simplicity of exposition, we assume that the time and space are continuous. There are two kinds of messages: (a) Broadcast messages: A broadcast message originating at a position P travels in concentric hyperspheres centered at P in all directions, (b) Directional messages: A ideal directional message travels only in a specific direction specified by a ray. Such messages can be sent by lasers or directional antennas.

We assume that all the players have to use two kinds of ideal devices to communicate with each other: atomic sending devices and atomic receiving devices. Before defining these ideal devices, we define the *Send* and *Receive* oracles. The oracles simulate the sending/receiveing actions in the real world.

We assume there is a message pool with infinite capacity which can be accessed by *Send* and *Receive* only. There are k communication channels in the system.

- $Send(c, m, T, \boldsymbol{d})$. Denote the sender's own position by P. On inputs a channel index $c \in \mathbb{Z}_k$, one bit content $m \in \{0, 1\}$, a sending time T and a direction vector $\boldsymbol{d}$, the *Send* oracle generates a message $< c, m, T, P, \boldsymbol{d} >$ and adds it into the message pool. $\boldsymbol{d} = 0$ if it is a broadcast message. $\boldsymbol{d}$ is a unit vector $|\boldsymbol{d}| = 1$ if it is a directional message.
- $Receive(c', T')$. Denote the receiver's own position by P'. On inputs a channel index c', a receiving time T', if there is only one message $< c, m, T, P, \boldsymbol{d} >$ in the message pool which satisfy that
  - the receiving channel is as same as the sending channel $c' = c$; and
  - it is a broadcast message $\boldsymbol{d} = 0$ passing through P' at time T'

$$\frac{|\overrightarrow{PP'}|}{c} = T' - T$$

or a directional message $\boldsymbol{d} \neq 0$ passing through P' at time T'

$$\frac{\overrightarrow{PP'}}{c\boldsymbol{d}} = T' - T$$

the *Receive* oracle outputs the content m; otherwise it outputs a random bit.

Spoofing one's own position while calling the oracle is not allowed because it is impossible in the real world. A device cannot send a message from position A while it is indeed at position B. Thus, the caller's position is not an input to the oracles. We assume that the oracles have already known it.

Next, we define the atomic sending device and the atomic receiving device.

- **Atomic sending device.** A atomic sending device enable its owner to call the *Send* oracle once per time slot.
- **Atomic receiving device.** A atomic receiving device enable its owner to call the *Receive* oracle once per time slot.

Thus, a player which have m atomic sending/receiving devoices can call the *Send*/*Receive* orale m times in one time slot. By constraining the maximum number of atomic devices that each party maintains, we define the sub-models.

- **Sending-constrained multi-channel model**. A multi-channel is sending-constrained if for every party in the model, the number of atomic sending devices that they maintain is at most q_s.
- **Receiving-constrained multi-channel model**. A multi-channel is receiving-constrained if for every party in the model, the number of atomic receiving devices that they maintain is at most q_r.
- **Both-constrained multi-channel model**. A multi-channel is both-constrained if for every party in the model, the number of atomic sending devices that they maintain is at most q_s and the number of atomic receiving devices that they maintain is at most q_r.

2.2 Position-Verification

Next, we define a general position-verification protocol in the MCM. We refer to the definition in [BCF+10]. Adapting to the MCM, our definition is quite different from theirs although the essences are similar.

There are three roles of players in a position-verification protocol: prover, verifier and adversary (dishonest prover). Players in the same role buildup a party. We assume that the verifiers are honest. They will never share any private information with the adversaries voluntarily.

A position-verification protocol starts when a set of n verifiers $V_1, ..., V_n$ which are located at $pos_1, ..., pos_n$ get a verification request from a prover or a adversary which claims that it is at position pos. After executing the protocol, the verifiers accept if they believe that someone is indeed at pos.

Definition 1. *A **position-verification protocol** PV consists of a set of verifiers' challenge algorithms $VER = \{Ver_1 ... Ver_n\}$ and a prover response algorithm Pro. Each challenge algorithm $Ver_i(req, pos, pos_1, ..., pos_n) \in \{0, 1\}$ on inputs the verification request req, the prover's position and the verifiers' positions outputs the verification result. The prover response algorithm $Pro(req, pos, pos_1, ..., pos_n)$ responds to the verifiers' challenges.*

Here, the inputs and the outputs of the algorithms mean local inputs and outputs. We treat the communication between the players by calling the *Send*/*Receive* oracles as parts of the algorithms. We assume that for each execution, the outputs of all the verifiers' challenge algorithms $Ver_1 ... Ver_n$ are the same. To ensure that, the verifiers may exchange their verification results before output. Thus, w.l.o.g. we take the output of Ver_1 as the verification result of all the verifiers.

Definition 2. *A position-verification protocol PV has* ***perfect completeness*** *if for a honest prover at a verification-available position pos running Pro, the verifiers accept* $Ver_1 = 1$ *with probability 1 .*

Definition 3. *A position-verification protocol PV is* ε-***sound*** *if for any verification-available position pos, any coalition of dishonest provers* $\hat{P}_1, ... \hat{P}_l$ *at arbitrary positions* $apos_1, ... apos_l, all \neq pos$, *the verifiers accept* $Ver_1 = 1$ *with probability at most* ε.

3 Our Protocol

In this section, we propose a 1-round position-verification protocol and calculate its soundness under different constraints.

In the protocol, we need only two verifiers in any space of arbitrary dimensions. Denote the two verifies by V_1 and V_2. Denote the honest prover by P. Let the angle between line PV_1 and line PV_2 be α. We require that $\pi/2 \leq \alpha < \pi$.

All the messages in this protocol are directional messages. Switching to a certain channel and sending/receiving a message can be done instantaneously. This assumption can be removed by making an acceptable uncertainty in distance. The further details can be found in Section 4.2.

Let $CH = \{0, 1, ...k-1\}$ be the set of all the channels in the MCM. We assume that the verifiers are able to send $\lceil \log k \rceil$ bits at the same time and the prover is able to receive $\lceil \log k \rceil$ bits at the same time. That means they have $\lceil \log k \rceil$ atomic sending/receiving devices. The number of the atomic devices that the honest players need can be reduced by making an uncertainty in distance. The further details can be found in Section 4.3.

Every two verifiers have a pre-shared key which allows them to exchange private messages. The adversaries can send private messages to each other, too. In addition, the verifiers need a pseudorandom generator to generate secure random numbers.

The prover's verification request consists of two parts: its position claim $pos \in \mathbb{R}^d$ (d-dimensional space) and a time window $[T_1, T_2]$. The time window shows when the prover P is at *pos* so that the challenge messages should arrive at P between T_1 and T_2. We assume that the length of the time slot in the system is t_s and $T_2 - T_1$ is divisible by t_s. Let $t_w = T_2 - T_1$ and $l = t_w/t_s$. Let t_1 and t_2 be the time taken for radio waves to reach P from V_1 and V_2 respectively. We require that $t_w > t_1 + t_2 - \sqrt{t_1^2 + t_2^2}$.

Before verification, V_1 runs the secure pseudorandom generator to prepare a set of $\lceil \log k \rceil$ public channels $PUB = \{pub_1, ...pub_{\lceil \log k \rceil}\}, pub_i \leftarrow_R CH$. V_1 sends PUB to the prover publicly. V_1 also generates a secret channel $sec \leftarrow_R CH \setminus PUB$, a challenge bit $m \leftarrow_R \{0, 1\}$ and a meeting time $T = T_1 + xt_s, x \leftarrow_R \{0, ..., l-1\}$ which is the time that the messages from V_1 and V_2 should meet at P. V_1 sends sec, m, T to V_2 privately by using the pre-shared keys.

Denote i-th bit of *sec* by sec_i. The protocol is show in Figure 1.

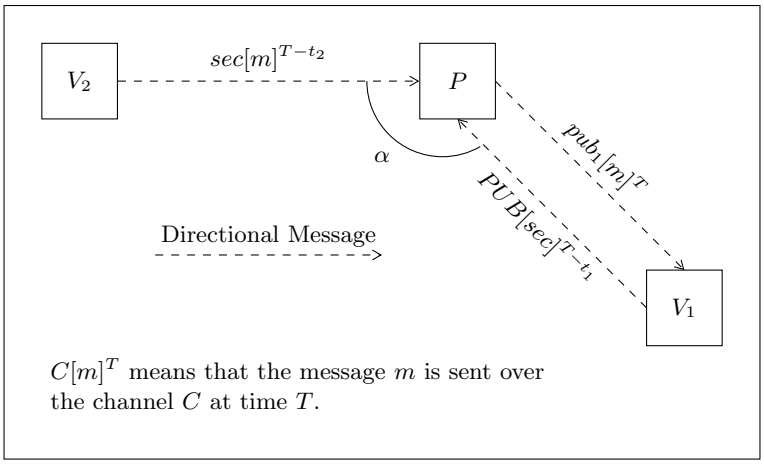

Fig. 1. 1-round Position Verification protocol

1. At time $T - t_1$, V_1 sends sec to P over the public channels PUB. To accomplish that, V_1 calls

$$Send(pub_i, sec_i, T - t_1, \frac{\overrightarrow{V_1P}}{|\overrightarrow{V_1P}|}), i = 1, ..., \lceil \log k \rceil$$

   oracle for $\lceil \log k \rceil$ times.
2. At time $T - t_2$, V_2 calls

$$Send(sec, m, T - t_2, \frac{\overrightarrow{V_2P}}{|\overrightarrow{V_2P}|})$$

   to send m over the channel sec.
3. From T_1 to T_2, in each time slot $j = 0, ..., l - 1$, the prover calls

$$Receive(pub_i, T_1 + jt_s) = sec'_{ji}, i = 1, ..., \lceil \log k \rceil$$

   to get sec'_j. Then, it calls

$$Receive(sec'_j, T_1 + jt_s) = m'_j$$

   to get m'_j. Immediately, it calls

$$Send(pub_1, m'_j, T_1 + jt_s, \frac{\overrightarrow{PV_1}}{|\overrightarrow{PV_1}|})$$

   to send the challenge bit m'_j back to V_1 over the channel pub_1. Note that at time $T = T_1 + xt_s$, the prover sends the correct challenge bit back $m'_x = m$. [1]
4. At time $T + t_1$, V_1 calls

$$Receive(pub_1, T + t_1) = m''$$

   to obtain m''. If $m = m''$, the verifiers accept the position claim. Otherwise, they reject the position claim.

It is not hard to check that our 1-round protocol has perfect completeness. Not that by repeating a ε-sound 1-round position-verification protocol n times, we directly have a ε^n-sound position-verification protocol. Since there is no computation in our protocol, we evaluate the efficiency by the communication complexity. The number of communication actions in our 1-round position-verification protocol is $O(\log k)$. If we repeat the 1-round protocol n times, the total number of the communication actions is $O(n \log k)$.

Next, we discuss the security of the 1-round protocol.

3.1 Security in the rc-MCM

In the rc-MCM, the receiving ability of each party is constrained by the maximum number of their atomic receiving devices q_r. As the total number of channels k is larger than q_r, the challenge message in a random channel slips by the adversaries with high probability while they do not know the channel beforehand.

We assume that the atomic sending devices of adversaries are deployed before the execution of the protocol and will not move during the protocol. This is reasonable because compared with the speed of light the physical movements of the devices are negligible.

As the challenge bit m is sent by a directional message on the ray V_2P, the adversaries which can tap or relay m must locate on the ray V_2P. For example, the adversary A_3 in Figure 2 cannot gather any information about m. Next, we prove that the adversaries obtain m with negligible probability, even if they put all of their atomic receiving devices on the ray V_2P.

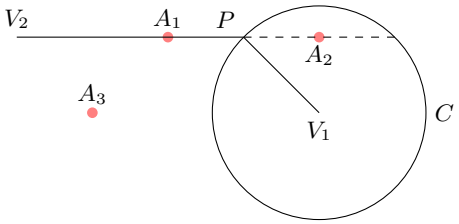

Fig. 2. Security in the MCM

Lemma 1. *In the rc-MCM, any coalition of dishonest provers (adversaries) $\hat{P}_1, ...\hat{P}_l$ at arbitrary positions not equal to pos can obtain the challenge message m in time with the probability at most $\frac{q_r}{k}$. Here, in time means the adversaries have enough time to send the obtained m to V_1.*

Proof. Since the channel number *sec* is sent by V_1 in public channels, any one on the path of this message can broadcast it. Assume that an adversary is very close to V_2. When the adversary obtains *sec*, it broadcasts the messages. Therefore, the knowledge of *sec* spreads at the speed of light as concentric hyperspheres centered at V_1. For example, at time T, all the players in the circle C in Figure 2 know *sec* but no player out of C knows *sec*.

On the other hand, the challenge message m travels on the ray V_2P. If m has not been relayed on the line segment V_2P, as $\alpha \geq \frac{\pi}{2}$, the first meeting place of the knowledge of *sec* and the challenge

[1] The correct meeting time T is unknown to the prover. We show how to reduce the number of sending actions of the prover in Section 4.4.

message m is P. The adversaries after P on the path of m (e.g. A_2 in Figure 2) can obtain m because they can know *sec* before m's arrival, however, they cannot send m back to V_1 in time.

Thus, we only need to calculate the probability that m is obtained by the adversaries on the line segment V_2P. The adversaries on the line segment V_2P (e.g. A_1 in Figure 2) cannot obtain *sec* before the arrival of m. The only way to get m is to first guess *sec*, which succeeds with probability at most $\frac{q_r}{k}$. Note that if the adversaries on the line segment V_2P obtain m, they can send m back to V_1 in time.

Thus, the probability that the adversaries obtain the challenge message m in time is at most $\frac{q_r}{k}$.

Theorem 1. *Our position-verification protocol is* $(\frac{1}{2} + \frac{q_r}{2k})$*-sound in the rc-MCM.*

Proof. For any coalition of dishonest provers $\hat{P}_1, ...\hat{P}_l$ at arbitrary positions not equal to *pos*, if they can obtain the challenge bit in time, the verifiers accept with the probability 1, otherwise, they merely guess the challenge bit, the verifiers accept with the probability $\frac{1}{2}$. By applying Lemma 1, the verifiers accept with the probability

$$\frac{q_r}{k} \times 1 + \left(1 - \frac{q_r}{k}\right) \times \frac{1}{2} = \frac{1}{2} + \frac{q_r}{2k}$$

3.2 Security in the sc-MCM

In the receiving constrained model, the adversaries cannot send the challenge bit back in time because they obtain the bit with negligible probability. In the sending constrained model, they can obtain the challenge bit. There is no constraint on their receiving ability. However, without the channel number and the exact sending time of the challenge bit, they cannot recognize the challenge bit from the random bits in all the channels. As their sending abilities are constrained by the maximum number of their atomic sending devices q_s, they cannot send all these random bits to their partners quickly enough.

As the challenge bit m is sent by a directional message on the ray V_2P, the adversaries which can tap or relay m must located on the ray V_2P. Next, we prove that the adversaries relay m in time with low probability, even if they put all of their atomic sending devices on the ray V_2P.

Lemma 2. *In the sc-MCM, any coalition of dishonest provers (adversaries)* $\hat{P}_1, ...\hat{P}_l$ *at arbitrary positions not equal to pos can relay the challenge message* m *in time with the probability at most* $\frac{2q_s}{k}$*. Here, in time means the adversaries have enough time to send the relayed* m *to* V_1*.*

Proof. Since the channel number *sec* is sent by V_1 in public channels, any one on the path of this message can broadcast it. Assume that an adversary is very close to V_2. When the adversary obtains *sec*, it broadcasts the messages. Therefore, the knowledge of *sec* spreads at the speed of light c as concentric hyperspheres centered at V_1.

On the other hand, the challenge message m travels on the ray V_2P. If m has not been relayed on the line segment V_2P, as $\alpha \geq \frac{\pi}{2}$, the first meeting place of the knowledge of *sec* and the challenge message m is P. The adversaries after P on the path of m (e.g. A_2 in Figure 2) cannot send m back to V_1 in time.

Thus, we only need to calculate the probability that m is relayed by the adversaries on the line segment V_2P. However, the adversaries on the line segment V_2P (e.g. A_1 in Figure 2) cannot obtain

sec before the arrival of m and they do not know the exact time when m passes them. Thus, they cannot recognize m. They just try to relay as many bits as possible.

Assume that there is an adversary A_1 at a place on V_1P and very close to V_1 which is waiting for the relayed bits. If m is in the relayed bits from the adversaries on the line segment PV_2, A_1 can recognize it because A_1 have known *sec* already.

The best place of an adversary A_2 at which they have the most relay time is at a place very near V_2 on the ray V_2P. If A_2 relays m to A_1 before $T + t_1 - \sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha}$, A_1 can send m to V_1 on time and win the game. Here, $\sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha}$ is the transmitting time of a message from A_2 to A_1 directly. The relay time that A_2 has is $(T_2 + t_1 - \sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha}) - (T_1 - t_2)$. $(T_1 - t_2)$ is the first time point that the challenge bit m is possibly in the bits passing A_2 if V_2 sends m at $T_1 - t_2$. $(T_2 + t_1 - \sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha})$ is the last time point that m can be relayed to A_1 in time if V_2 sends the challenge bit m at $T_2 - t_2$, .

In such time, the bits that A_2 is able to relay is at most

$$q_s \frac{T_2 - T_1 + t_2 + t_1 - \sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha}}{t_s}$$

The challenge bit may be sent by V_2 in any time slot of the time window $[T_1 - t_2, T_2 - t_2]$ over any channel of the k channels. The total amount of the bits that may be the challenge bit is

$$k\frac{T_2 - T_1}{t_s}$$

Note that $-1 < \cos\alpha \leq 0$ since $\frac{\pi}{2} \leq \alpha < \pi$ and $t_w > t_1 + t_2 - \sqrt{t_1^2 + t_2^2}$.

Thus, m is in the relayed bits with probability at most

$$\frac{q_s(T_2 - T_1 + t_2 + t_1 - \sqrt{t_2^2 + t_1^2 - 2t_2t_1\cos\alpha})}{k(T_2 - T_1)} \leq \frac{q_s(t_w + t_2 + t_1 - \sqrt{t_2^2 + t_1^2})}{kt_w} < \frac{2q_s}{k}$$

Thus, the probability that the adversaries relay the challenge message m in time is at most $\frac{2q_s}{k}$.

Theorem 2. *Our position-verification protocol is* $(\frac{1}{2} + \frac{q_s}{k})$*-sound in the sc-MCM.*

Proof. For any coalition of dishonest provers $\hat{P}_1, ...\hat{P}_l$ at arbitrary positions not equal to *pos*, if they can relay the challenge bit in time, the verifiers accept with the probability 1, otherwise, they merely guess the challenge bit, the verifiers accept with the probability $\frac{1}{2}$. By applying Lemma 2, the verifiers accept with the probability

$$\frac{2q_s}{k} \times 1 + \left(1 - \frac{2q_s}{k}\right) \times \frac{1}{2} = \frac{1}{2} + \frac{q_s}{k}$$

3.3 Security in the bc-MCM

In most real scenarios, both of the user's sending ability and receiving ability are constrained.

Theorem 3. *Our position-verification protocol is* $(\frac{1}{2} + \min\{\frac{q_r}{2k}, \frac{q_s}{k}\})$*-sound in the bc-MCM.*

Proof. The bc-MCM is a special case of the rc-MCM. By applying Theorem 1, our protocol is $(\frac{1}{2} + \frac{q_r}{2k})$-sound. And, the bc-MCM is a special case of the sc-MCM. By applying Theorem 2, our protocol is $(\frac{1}{2} + \frac{q_s}{k})$-sound. Thus, our protocol is $(\frac{1}{2} + \min\{\frac{q_r}{2k}, \frac{q_s}{k}\})$-sound in the bc-MCM.

4 Discussions

4.1 Relation with Some Leakage Models

We notice that our sub-models of the MCM have some similar properties with the leakage models used in leakage-resilient cryptography.

The first provablly secure position-verification protocol proposed by Chandran et al. [CGMO09] is based on the BRM. The BRM is a leakage model. It is based on an assumption, that adversaries can only retrieve part information of a long random string. The BRM is a widely used model [CLW06,DP07,ADW09]. Our rc-MCM can be considered as a specialized BRM. If we consider all the bits over k channels in the rc-MCM as the long random string in the BRM, the bits which can be received by q_r atomic receiving devices of the adversaries in the rc-MCM are the retrievable part of the long string in the BRM. Thus, most of the protocols in [CGMO09] can be translated into the rc-MCM. In [CGMO09], it requires the verifiers to be able to broadcast large bursts of information in different channels to send out the long random strings. That is a great harm to the efficiency of their protocols. However, in the MCM, we do not need such long strings. Thus, the efficiency of their protocols can be promoted by adopting the MCM. We show a example in Appendix A.

Another model called the bounded storage model (BSM) is much related to the BRM. It is based on an assumption that there is a bound on the amount of information that each party can store. This is a reasonable assumption in the past, although artificially making this assumption in practice is a little expensive nowadays. The BSM was first introduced by Maurer [Mau92]. This model has been the subject of much work [Mau92,CM97,CCM98,AR99,Din01,ADR02,DR02,Lu04,Vad04,DM04a] [DM04b,Din05,DHRS07,MSTS09].

The adversaries in our sending-constrained model have much more powers than that in the BRM and BSM. The adversaries in sc-MCM can obtain and store all the bits over the k channels. Thus, the position-verification protocols with only broadcast messages are not secure in sc-MCM. By the triangle inequality, an adversary located very close to a verifier can obtain all the broadcast messages in time so that it can send the challenge messages back to the verifier on time. However, the adversaries in sc-MCM have troubles on sharing their knowledge with each other quickly since their sending abilities are constrained. If an adversary wants to share a directional message in an unknown channel with its partner which is not on the path of the message, the adversaries have to relay the messages in all the channels to change the path, that is hard to do in the sc-MCM. That is why our protocol can be secure in the sc-MCM.

4.2 Remove the Instantaneous Actions

In the above discussions, we assume that the receiving and sending actions in the system are instantaneous for clarity. Using the same technique as [CGMO09], we can remove the assumption. In this section, we mainly introduce the technique.

The brief idea is to add some necessary delays into the protocols. For example, in the original protocol, the verifiers send two messages *sec* and m which reach the prover at the "same" time and the prover needs to receive *sec* before m. In the compiled protocol, the verifiers should send *sec* at first. After a *standstill* for time t, the verifiers send m out. This means that the verifiers allow the prover to receive *sec* in time t.

In previous position-based protocols, researchers use the above technique to fill the time slot caused by the computations and the sending/receiving actions. The authors in [CHH09] point

that the uncertainties in distance caused by the computations are unacceptable. This is because an adversary with stronger computation power can shorten the computation time to attack the protocol. For example, if an adversary can do a certain computation $1\mu s$ faster than an honest player, it may compute some confidential information *on time* at a place about $300m$ away from the place where the honest player should be. *Fortunately, in our protocol, there is no computation but only sending and receiving actions.*

4.3 Reduce the Parallel Actions

In our protocol, we require that the verifiers are able to send $\lceil \log k \rceil$ bits at the same time, and the prover is able to receive $\lceil \log k \rceil$ bits at the same time. This might cost too much in some scenarios. Here, we show how to reduce the parallel actions by making an uncertainty in distance.

The method is to replace these parallel actions by serial actions. In the original protocol, V_1 sends the $\lceil \log k \rceil$ bits of *sec* at the same time. Now, we change it as that V_1 sends these bits one by one. Correspondingly, the prover does not have to receive the bits concurrently any more. It can receive them serially.

This change makes that the adversaries near P have better probability to make a successful attack. See Figure 3. We draw an arc centered at V_1 with radius $|V_1P| + ct_s\lceil \log k \rceil$ which intersects V_2P at P'. We assume that *sec* is broadcasted by someone very near V_1 again. Thus, players on the segment PP' more or less know some bit of *sec* while the challenge bit m passing them. For example, the adversary at P' knows one bit of *sec* and the honest prover at P knows all the bits of *sec*. The closer to P, the more bits of *sec* it obtains. That makes them obtain m more easily.

In the rc-MCM, if an adversary is placed on the segment PP', it can obtain t bits of *sec* beforehand. Then, the adversaries can obtain m in time with the probability $\frac{2^t q_r}{k}$. Similarly, in the sc-MCM, the adversaries can relay m in time with the probability $\frac{2^{t+1} q_r}{k}$.

To deal with the security loss, a simple way is just assume that no adversary is on the segment PP'. If the bit rate of the system is high enough, PP' can be quite short. This becomes a tradeoff. If you want higher security, deploy more communication devices in the system and use parallel actions. If you want lower cost, use the serial actions and hope no adversary is in the forbidden region.

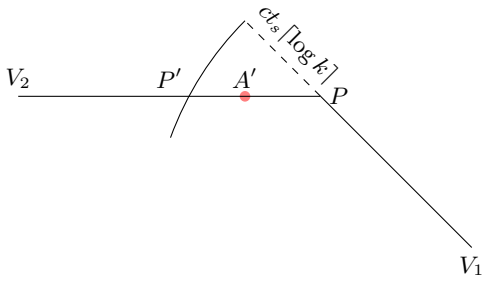

Fig. 3. Security analysis of the changed protocol

4.4 Reduce the Sending Actions of the Prover

The prover in our protocol need to send a bit to V_1 per time slot in the time window $[T_1, T_2]$. Sending actions are more costly than receiving actions in wireless setting. In practice, the prover is probably a mobile device. Too many sending actions would consume its batteries soon. On the other hand, the verifiers are probably infrastructures. More sending actions to them might be okay. Next, we hand out a method to reduce the sending actions of the prover.

The brief idea is that V_1 sends a special tag along with sec to the prover. Let PUB' be another set of public channels that are all different from the channels in PUB. The verifiers tell PUB' to the prover just like PUB. At $T - t_1$, V_1 sends the special tag (e.g. all zero bits) in PUB' and sec in PUB. In the time window $[T_1, T_2]$, the prove continually listens to the channels of PUB'. If sometime the bits in PUB' equals to the special tag, the prover sends the received m' in that time slot to V_1. Thus, the number of the sending actions of the prover becomes $\frac{1}{2^{l_t}}$ as many as that in the original protocol. Here, l_t is the length of the special tag.

However, this is a tradeoff, too. The verifier V_1 needs to send more bits in the new protocol. And, the number of the parallel actions increases that means the prover and the verifiers need more atomic communication devices. Of course, we can use the method in Section 4.3 to reduce the parallel actions by sacrificing part of the security.

4.5 Improve the Security

If the constraints in the MCM are not tight enough, the soundness of our 1-round protocol becomes 1 (not secure at all), for example, in the bc-MCM, if $q_r, q_s > k$. That means the adversaries can make an ideal mirror which is able to reflect everything.

Next, we adopt a notion called covert verifier to improve the security in this setting.

This idea comes from [CCS06]. They proposed several position-verification protocols based on the covert verifiers(covert base station in their paper). Such verifiers' position is unknown to the adversaries before the execution of the protocols. In [CGMO09], some attacks to these protocols are given which shows that the security is weak if we only rely on the covert verifiers.

We find that if the constraints are not greatly larger than the channel number k , such that the adversaries cannot build up too many ideal mirrors, by adopting covert verifiers, the soundness of our protocol can still be significantly smaller than 1.

We assume that there are lots of verifier candidates in the system and they are moving randomly. Vehicles in the vehicle network (VANET) or mobile phones in the mobile phone network can be such verifier candidates.

In the new protocol, the verification request is handled by a verification center. The center randomly picks two appropriate verifier candidates to be the verifiers. The center tell the position of V_1 (that is the destination of the response message) to the prover (or the adversaries), and keep the position of V_2 as a secret. The following steps run as same as that in the original protocol.

The security analysis of the new protocol is similar to that of the original protocol. The difference is that since the adversaries do not know the position of V_2, they have to deploy their atomic communication devices on every possible directions. Thus, the probability of that too many atomic communication devices (k in the rc-MCM and $k/2$ in the sc-MCM) are deployed by the adversaries on the segment V_2P is less then 1. Then, the soundness of the new protocol is still smaller than 1.

However, our protocol is not secure if the adversaries have a huge number of atomic communication devices such that they can deploy k atomic receiving devices and $k/2$ atomic sending devices

on every possible path of the challenge bit. This confirms that it is impossible to propose a secure position-verification protocol without any constraint.

4.6 Combine with Other Cryptosystems

In [BCF+10], a general general method to propose a position-based key-exchange protocol from a ε-sound 1-round position-verification protocol is given. Based on the position-based key-exchange protocol, various position-based tasks can be fulfilled.

The positive side of this construction is that the privacies of the users (provers) is well protected. The identity of a user in this system is its geography position, moreover, it is the only information of the user that is known to others. The negative side is that the construction is not efficient enough. A component called position-based authentication is needed. However, the only existing position-based authentication protocol proposed in [BCF+10] is not efficient enough.

Here, we give an alternative approach. We sacrifice part of the privacy of the users to improve the efficient. The approach is quite straightforward. We combine our position-verification protocol with any efficient cryptosystem which supports signature and mutual authentication. Firstly, the prover hands out a position claim with its signature on this claim and runs the position-verification protocol with the verifiers. If the position claim accepts, the prover and the verifiers use an arbitrary authenticated key-exchange protocol to share a key. The efficiency of this approach is guarantee by the efficiency of our position-verification protocol and the efficiency of the chosen cryptosystem. However, the verifiers know much more about the prover beside the position of the prover: its identity in the cryptosystem, its public key...

5 Conclusion

In this paper, we define a cryptographic model called MCM. The MCM is an abstraction from the scenarios in which there are lots of channels but each party can just control part of these channels at the same time. In the MCM, communication over a random channel can achieve privacy. We characterize the sending and receiving actions in the MCM and define three sub-models of the MCM with different constraints on the sending and receiving abilities of the players. Finally, we propose a position-verification protocol and prove its security in all these sub-models.

References

[ADR02] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In *CRYPTO*, pages 65–79, 1999.

[BCF+10] Harry Buhrman, Nishanth Chandran, Serge Fehr, Ran Gelles, Vipul Goyal, Rafail Ostrovsky, and Christian Schaffner. Position-based quantum cryptography: Impossibility and constructions. *CoRR*, abs/1009.2490, 2010.

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *FOCS*, pages 493–502, 1998.

[CCS06] Srdjan Capkun, Mario Cagalj, and Mani B. Srivastava. Secure localization with hidden and mobile base stations. In *INFOCOM*, 2006.

[CGMO09] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position based cryptography. In *CRYPTO*, pages 391–407, 2009.

[CH05] Srdjan Capkun and Jean-Pierre Hubaux. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM*, pages 1917–1928, 2005.

[CHH09] Jerry T. Chiang, Jason J. Haas, and Yih-Chun Hu. Secure and precise location verification using distance bounding and simultaneous multilateration. In *WISEC*, pages 181–192, 2009.

[CLW06] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.

[CM97] Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In *CRYPTO*, pages 292–306, 1997.

[DHRS07] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. *J. Cryptology*, 20(2):165–202, 2007.

[Din01] Yan Zong Ding. Oblivious transfer in the bounded storage model. In *CRYPTO*, pages 155–170, 2001.

[Din05] Yan Zong Ding. Error correction in the bounded storage model. In *TCC*, pages 578–599, 2005.

[DM04a] Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In *EUROCRYPT*, pages 126–137, 2004.

[DM04b] Stefan Dziembowski and Ueli M. Maurer. Optimal randomizer efficiency in the bounded-storage model. *J. Cryptology*, 17(1):5–26, 2004.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237, 2007.

[DR02] Yan Zong Ding and Michael O. Rabin. Hyper-encryption and everlasting security. In *STACS*, pages 1–26, 2002.

[Lu04] Chi-Jen Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *J. Cryptology*, 17(1):27–42, 2004.

[Mau92] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *J. Cryptology*, 5(1):53–66, 1992.

[MSTS09] Tal Moran, Ronen Shaltiel, and Amnon Ta-Shma. Non-interactive timestamping in the bounded-storage model. *J. Cryptology*, 22(2):189–226, 2009.

[SP05] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 7pp.–840, Nov. 2005.

[SSW03] Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *Workshop on Wireless Security*, pages 1–10, 2003.

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.

[VN06] Adnan Vora and Mikhail Nesterenko. Secure location verification using radio broadcast. *IEEE Trans. Dependable Sec. Comput.*, 3(4):377–385, 2006.

[ZLFW06] Yanchao Zhang, Wei Liu, Yuguang Fang, and Dapeng Wu. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(4):829–835, 2006.

A Translate a Protocol from the BRM into the MCM

Here, we translate the 1-dimensional position-verification (secure positioning) protocol in [CGMO09] into the MCM. We point out that with the similar method, most protocols in [CGMO09] can be translated into the MCM. The protocol is shown in Fig. 4.

Let $CH = \{0, 1, ...k\}$ be the set of all the channels in the MCM. We assume that the prover and the verifiers are able to send/receive $\lceil \log k \rceil$ bits at the same time. That means they all have $\lceil \log k \rceil$ atomic sending/receivng devices.

The prover's verification request consists of two parts: its position claim $pos \in \mathbb{R}$ (1-dimensional space) and a time T. T is the time when the challenge messages should arrive at P. Let t_1 and t_2 be the time taken for radio waves to reach P from V_1 and V_2 respectively.

Before verification, V_2 runs the secure pseudorandom generator to prepare a set of $\lceil \log k \rceil$ public channels $PUB = pub_1, ...pub_{\lceil \log k \rceil}, pub_i \leftarrow_R CH$. V_2 sends PUB to the prover publicly. V_2 also generates a secret channel $sec \leftarrow_R CH$ and a challenge bit $m \leftarrow_R \{0, 1\}$. V_2 sends sec, m to V_1 privately by using the pre-shared keys.

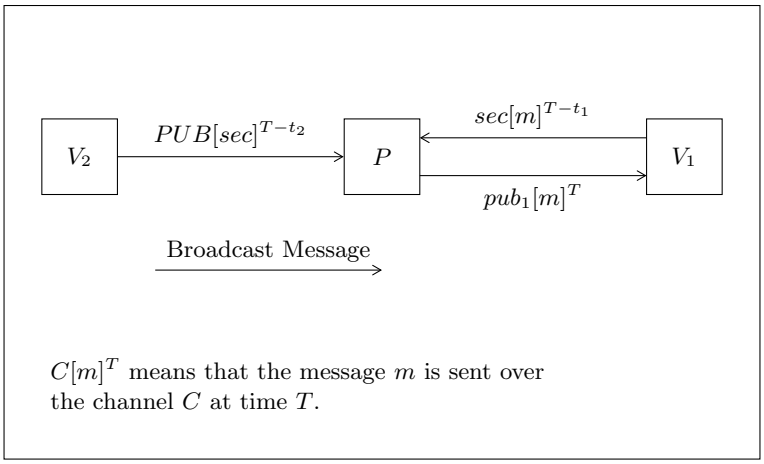

Fig. 4. 1-dimensional position-verification protocol

1. At time $T - t_2$, V_2 sends sec to P over the public channels PUB. To accomplish that, V_2 calls
$$Send(pub_i, sec_i, T - t_2, 0), i = 1, ..., \lceil \log k \rceil$$
oracle for $\lceil \log k \rceil$ times.
2. At time $T - t_1$, V_1 calls
$$Send(sec, m, T - T_1, 0)$$
to send m over the channel sec.
3. At time T, the prover calls
$$Receive(pub_i, T) = sec'_i, i = 1, ..., \lceil \log k \rceil$$
to get sec'. Then, it calls
$$Receive(sec', T) = m'$$
to get m'. Immediately, it calls
$$Send(pub_1, m', T, 0)$$
to send the challenge bit m' back to V_1 over the channel pub_1.
4. At time $T + t_1$, V_1 calls
$$Receive(pub_1, T + t_1) = m''$$
to obtain m''. If $m = m''$, it accept the position claim. Otherwise, it reject the position claim.

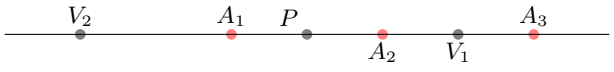

Fig. 5. Security of the 1-dimensional protocol

This protocol is secure in the rc-MCM but not secure in the sc-MCM. As this is a 1-dimensional protocol, we only need to consider three kinds of adversaries on the line V_2PV_1: the adversaries on

the ray PV_2 (A_1 in Figure 5), the adversaries on the line segment PV_1 (A_2 in Figure 5) and the adversaries on the ray PV_1 excluding the segment PV_1 (A_3 in Figure 5). A_1 and A_3 cannot send the challenge bit back to V_1 in time. Thus, we focus on the adversaries like A_2. In the rc-MCM, A_2 cannot obtain sec beforehand while $sec[m]$ passing it so that it obtain m with the probability at most $\frac{q_r}{k}$. Thus, this is a $(\frac{1}{2} + \frac{q_r}{2k})$-sound protocol in the rc-MCM.

However, in the sc-MCM, there is no constraint on the receiving abilities of A_2. A_2 can receive and store all the k bits over k channels with m. When the message $PUB[sec]$ comes, A_2 obtains sec and finds the correct bit of m in its storage, then, sends m to V_1 immediately. Thus, A_2 wins the game that means this protocol is not secure in the sc-MCM.