1-1-2013

A secure elliptic curve based RFID ownership transfer scheme with controlled delegation

Shu Cheng
Macquarie University, sc903@uow.edu.au

Vijay Varadharajan
Macquarie University

Yi Mu
University of Wollongong, ymu@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

A secure elliptic curve based RFID ownership transfer scheme with controlled delegation

Abstract

In practical applications, the owner of an RFID-tagged item canchange. In this paper, we propose a new RFID ownership transfer protocol using elliptic-curve cryptography. The paper first considers security and privacy requirements in the ownership transfer process. Then the paper provides a detailed description of our ownership transfer scheme outlining various protocol phases. Key features of the proposed scheme are that it allows controlled delegation and authorisation recovery, and the ownership transfer is achieved without a trusted third party. We describe a security analysis of the proposed scheme and demonstrate that it meets the desired security and privacy requirements. We also illustrate the performance results and show that our scheme is feasible for lightweight RFID tags.

Keywords

controlled, secure, scheme, delegation, elliptic, ownership, transfer, curve, rfid

Disciplines

Engineering | Science and Technology Studies

Publication Details

Cheng, S., Varadharajan, V., Mu, Y. & Susilo, W. (2013). A secure elliptic curve based RFID ownership transfer scheme with controlled delegation. Cryptology and Information Security Series, 11 31-43.

A Secure Elliptic Curve based RFID Ownership Transfer Scheme with Controlled Delegation

Shu CHENG [a], Vijay VARADHARAJAN [a] Yi MU [b] and Willy SUSILO [b]
[a] *Advanced Cyber Security Research Centre*
Department of Computing, Faculty of Science
Macquarie University, Sydney, Australia
e-mail:{shu.cheng, vijay.varadharajan}@mq.edu.au
[b] *Centre for Computer and Information Security Research*
School of Computer Science and Software Engineering
University of Wollongong, Wollongong, Australia
e-mail:{ymu,wsusilo}@uow.edu.au

Abstract. In practical applications, the owner of an RFID-tagged item can change. In this paper, we propose a new RFID ownership transfer protocol using elliptic-curve cryptography. The paper first considers security and privacy requirements in the ownership transfer process. Then the paper provides a detailed description of our ownership transfer scheme outlining various protocol phases. Key features of the proposed scheme are that it allows controlled delegation and authorisation recovery, and the ownership transfer is achieved without a trusted third party. We describe a security analysis of the proposed scheme and demonstrate that it meets the desired security and privacy requirements. We also illustrate the performance results and show that our scheme is feasible for lightweight RFID tags.

Keywords. RFID, Ownership Transfer, Security, Privacy, Controlled Delegation

1. Introduction

Radio Frequency Identification (RFID) has been long considered as a substitute for barcodes and offers several highly attractive features. RFID technology is widely used in many applications in our daily life, such as supply chain, access control, automatic payment, animal tracking and electronic passports [6]. RFID tags usually have limited memories and weak computational capabilities due to inexpensive cost and easy deployment. Therefore, RFID systems are vulnerable to various critical security threats. Over the recent years, several security and privacy concerns have been raised in many research works [6].

Communications between RFID tags and readers are vulnerable to various attacks. A tag could be eavesdropped and manipulated illegally since the communication between reader and tag are often via insecure wireless channel. In addition, each RFID tag contains a unique piece of information which can be used to identify itself. An adver-

sary can trace or distinguish a tag from other tags if the unique information is captured during communication between the tag and the reader. On the other hand, passive tags usually have limited memory and low processing capacity, and hence strong security approaches are infeasible in practice. Therefore, they are not usually tamper-resistant and are vulnerable to compromise.

Ownership means that only the owner has access to the tag and is able to interact with it in the secure manner; hence the owner and the tag should be able to authenticate each other. However, in many applications, during the lifetime of a tagged item, the owner of the item may change several times. When the ownership transfers, the previous owner needs to pass the secure information to the new owner. As a result, both the previous and the new owner are able to authenticate and identify the tag, and moreover, communicate with it. This may cause a problem since the privacy of either the tag or the owners can be easily infringed. A number of papers have been published to provide proper solutions for tag ownership transfer [12,16,14,11,4,17,8,2,3,7,13]. Most of the schemes are based on symmetric-key cryptographic algorithms such as hash functions and pseudo-random number generators because of the simplicity compared to asymmetric-key cryptography. However, it has been shown that such schemes often result in scalability as well as security and/or privacy problems. Recent works show that it is feasible to employ elliptic curve cryptography on lightweight RFID tags [1,5,10]. In such schemes, it is assumed that the RFID tags are able to process modular additions, modular multiplications and elliptic curve scalar multiplications. We will be using this assumption and proposing a public key absed RFID ownership transfer scheme.

1.1. Contribution and Paper Organisation

In this paper, we propose a RFID ownership transfer scheme. To the best of our knowledge, this is the first elliptic-curve based ownership transfer protocol. We show that the proposed scheme is secure and private. Furthermore, it is able to achieve the features like controlled delegation and authorisation recovery. The rest of this paper is organised as follows. The ownership transfer protocols are reviewed in Section 2. In Section 3, we give the description of the preliminaries for our scheme and outline the security and privacy requirements. Our elliptic-curve based RFID ownership transfer protocol is proposed in Section 4. In Section 5, we provide the security analysis of our scheme. In Section 6, we show the performance results of our scheme. Section 7 concludes this paper.

2. Related Work

Molnar, Soppera and Wagner [12] introduced the concept of ownership transfer in 2005. They proposed an RFID pseudonym protocol based on pseudo-random function and shared secrets. Their scheme employed a trusted centre in a tree structure to manage the shared secret with the tag. All the readers need the assistance from the trusted centre to authenticate the tag because only the trusted centre is able to identify the tag. The trusted centre controls the access privilege according to the ownership policy of the tag. After the ownership is transferred, the previous owner is not allowed to access the tag. Since the tag uses a unique pseudonym to each query, it is impossible for the previous owner to identify the tag without the help of the trusted centre. The trusted centre can also delegate

a reader limited access to the tag by giving it a derived key. For each query made by the readers, the tag generates a pseudonym using pseudo-random function and the derived key to protect its real ID. The tag also maintains a counter to determine the number of queries. After the counter reaches the maximum value designated by the trusted centre, the delegation automatically expires. There are several similar schemes [16,14,4,8] that also employ a trusted third party to control the ownership transfer. However, all of them are based on symmetric cryptographic primitives. The shared secret between the tag and the reader will be revealed to the adversary if the tag is compromised. Moreover, the privacy of the tag and the owner cannot be guaranteed.

Besides the protocols based on trusted third party, there also exist several ownership tranfer schemes involing only tags and owners. Saito, Imamoto and Sakurai [16] presented an ownership transfer scheme without the trusted third party. Upon receiving the ownership from the previous owner, the new owner updates the secret shared with the tag. However, their scheme is built under a fairly strong assumption that it is difficult for the adversary to exploit the communication channel from the tag to the reader because the range of the channel is short. Song [17] proposed an ownership transfer protocol as well as a security property called authorisation recovery. The previous owner is able to recover the ownership and temporarily interact with the tag. This property is quite useful considering in an after-sales scenario, the seller may need to verify the product before providing a repair service. However, the protocol does not provide information and location privacy and an adversary can perform a denial-of-service attack by simply blocking and forging the second message in the protocol flow [15]. Also, [13] argued that Song's protocol changes the share secret to previous owner's key for authorisation recovery, which actually means sharing ownership causing the ownership of the tag to become unclear. In RFIDSec'11, Fernández-Mir, Trujillo-Rasua, Castellá-Roca and Domingo-Ferrer [3] introduced a novel ownership transfer protocol that provides controlled delegation without the need of a stored counter in the tag. The server maintains a table storing the hash chain of MAX size to identify a tag. However, this protocol is vulnerable to denial-of-service attack because an adversary can always block the update message $MAX + 1$ times. The time consumption and storage cost could become huge even though the system may set the value MAX to a relatively high value to prevent the attack.

3. Preliminaries

In this section, first we outline the system model and assumptions, and then describe the security and privacy requirements for our scheme.

3.1. System Model and Assumptions

Each owner has his/her own personal reader, which is securely connected to his/her own database. Therefore, we consider the reader and the database as an entity and refer it as the reader. This model removes the need for a trusted centre that is required in a centralised model to maintain the current and/or previous ownership of each tag.

As there can be different settings in the same system model, we make the following assumptions for our model.

- The manufacturer is trusted. The manufacturer creates items and attaches a tag to every single item. It also writes the initial state in every tag.
- A tag has a rewritable memory, and is able to perform lightweight cryptographic operations.
- A tag is vulnerable to compromise attacks. That is to say, an adversary can obtain the internal secrets of a tag.
- An owner is an entity who engages in the ownership transfer. Each owner has a reader to communicate with the target tag.
- An owner communicates with the target tag via insecure radio-frequency interface. However, the communication between two owners is assumed to be secure.
- The current owner has the full control over its tag.

3.2. Security and Privacy Requirements

We now define our RFID security model. Firstly, the adversary is assumed to have complete control over the communication channel between tag and reader. Namely, it can observe, modify and block all exchanged messages, and generate new messages. The potential threats against the RFID system are listed as follows.

- Replay Attack: An adversary maliciously repeats previous communications between a reader and a tag to perform a successful authentication.
- Man-in-the-Middle Attack: An adversary inserts, modifies or deletes messages sent between a reader and a tag without being detected.
- Denial-of-Service Attack (De-Synchronization Attack): An adversary blocks or tampers with messages passed on between a reader and a tag, which causes the reader and the tag to lose synchronisation so that they cannot authenticate each other in future communications.
- Backward Traceability: An active adversary is able to identify a target tag from the past interactions between the tag and a reader, using the knowledge of the tag's present internal state by corrupting the tag.
- Forward Traceability: An active adversary is unable to identify a target tag from the future interactions between the tag and a reader, using the knowledge of the tag's present internal state by corrupting the tag.

In addition to the potential threats against general RFID systems, we also identify the security requirements from the previous ownership transfer schemes.

- Previous Owner Privacy: When the ownership transfer protocol is completed, the new owner cannot trace past communications between the previous owner and the tag.
- New Owner Privacy: When the ownership transfer protocol is completed, the old owner cannot trace future communications between the new owner and the tag.
- Controlled Delegation: The present owner of the tag temporarily delegates the access right of the tag to another entity without giving out the ownership. The owner is able to cancel the delegation at anytime. Moreover, the delegation will automatically expires at some time.
- Authorisation Recovery: The previous owner is able to access the tag with permission granted by the current owner. The current owner can cancel the temporary

authorisation at anytime. This security property can be considered as a special case of controlled delegation.
The scheme that we are proposing in the paper aims to address the above security and privacy requirements.

4. Our Ownership Transfer Scheme

4.1. Setup

Let E be an elliptic curve defined over a field $\mathbb{Z}_p^*$, where p is an k-bit prime number. Assume the point P is a generator of $\mathbb{G}$, which is the group of points on the elliptic cure E. Let $(SK_{o_i}, PK_{o_i}) = (y_{o_i}, y_{o_i}P)$ be the public-private key pair of the ith Owner. Note that these key pair are used in tag-owner communications. The manufacturer is the special owner o_0. The manufacturer randomly chooses a public-private key pair $SK_t = x$, $PK_t = xP$ for the target tag when creating the product that the tag is attached to, and sets the internal state of the tag (SK_t, PK_{o_0}).

4.2. Protocol Phases

In this section we describe different phases of our scheme. Our scheme is composed of key change protocol, transfer protocol, key update protocol and controlled delegation protocol. The notations used in our scheme is illustrated in Table 1.

Table 1. Notations of the proposed scheme

Notation	Interpretation
SK_t	the private key of the tag
PK_t	the public key of the tag, also used as the tag's identity ID_t
PK_o	the public key of the current owner stored in the tag
SK_i	the private key of the ith owner
PK_i	the public key of the ith owner
PK_b	the backup public key stored in the tag
c	the counter stored in the tag
c_m	the maximun value of the counter for delegation
Auth(PK_o)	the tag authenticates the reader using the owner's public key

Key Change Protocol. In this protocol, the current owner (denoted by $\mathsf{Owner_i}$) updates its public key stored in the tag (denoted by Tag) with a temporary one so that the new owner (denoted by $\mathsf{Owner_{i+1}}$) will not be able to identify or trace the past interactions between the tag and the current owner after having the ownership of the tag. Prior to executing this protocol, we assume that the tag and the current owner both have each other's public key. We also assume that the owner has determined which tagged item that s/he wishes to transfer the ownership of. The protocol is depicted in Figure 1 and detailed as follows.

1. First, an $\mathsf{Owner_i}$ chooses two random number r_1, r_2 and sends $A_1 = r_1 P$, $A_2 = r_2 PK_t$ to Tag.
2. Tag generates a nonce r_t and answers with the following information: $M_1 = r_t PK_o$, $M_2 = SK_t^{-1} r_t A_2 + SK_t A_1$.
3. Upon receiving M_1 and M_2, $\mathsf{Owner_i}$ computes $r_1^{-1}(M_2 - r_2 SK_i^{-1} M_1)$ and checks whether the value equals ID_t. If not, $\mathsf{Owner_i}$ rejects the Tag and terminates the protocol execution; otherwise it randomly picks a temporary private key y'_o, computes $B_1 = (r_1 + r_2)P + SK_i^{-1} y'_o M_1$ and $B_2 = y'_o PK_t + SK_i^{-1} M_1$, and sends them to Tag. y'_o will be stored by $\mathsf{Owner_i}$ and passed to the new owner as the ownership in the future.
4. Tag computes $A_1 + SK_t^{-1} A_2 + r_t SK^{-1}(B_2 - r_t P)$ and checks whether the result equals to B_1. If so, it updates PK_o with the value $SK_t^{-1}(B_2 - r_t P)$; otherwise Tag terminates the protocol execution.

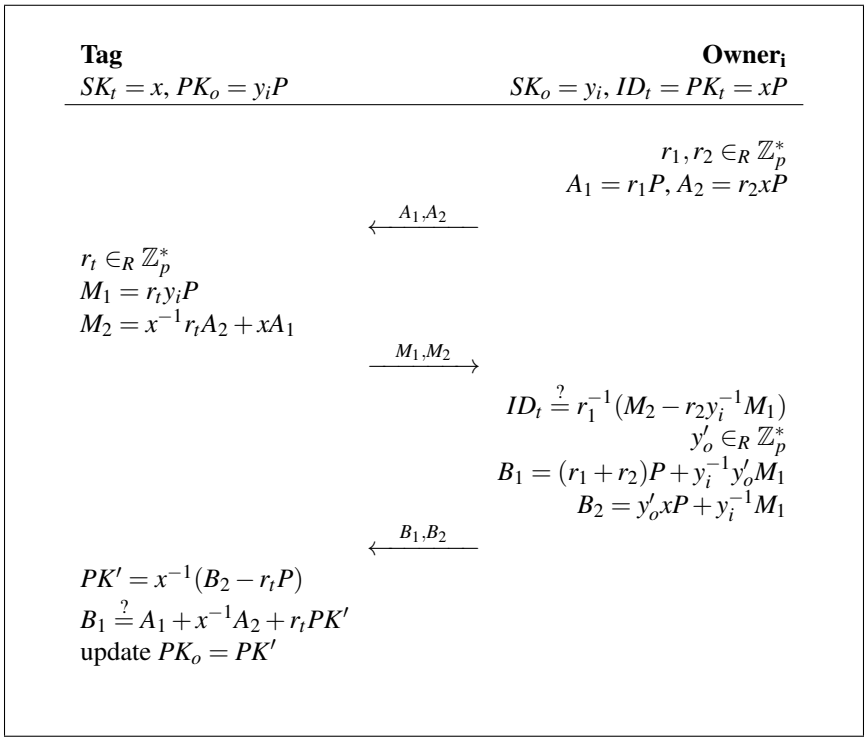

Figure 1. Key change protocol for current owner

Transfer Protocol. Since the interactions between owners are secure under the assumption, we assume the protocol is a general public-key based encryption protocol. The new owner encrypts the ownership transfer request and the ID of the tag and sends the message to the current owner. Then after decrypting the message and authenticating the new owner, the current owner encrypts the temporary private key y'_o for interacting with the

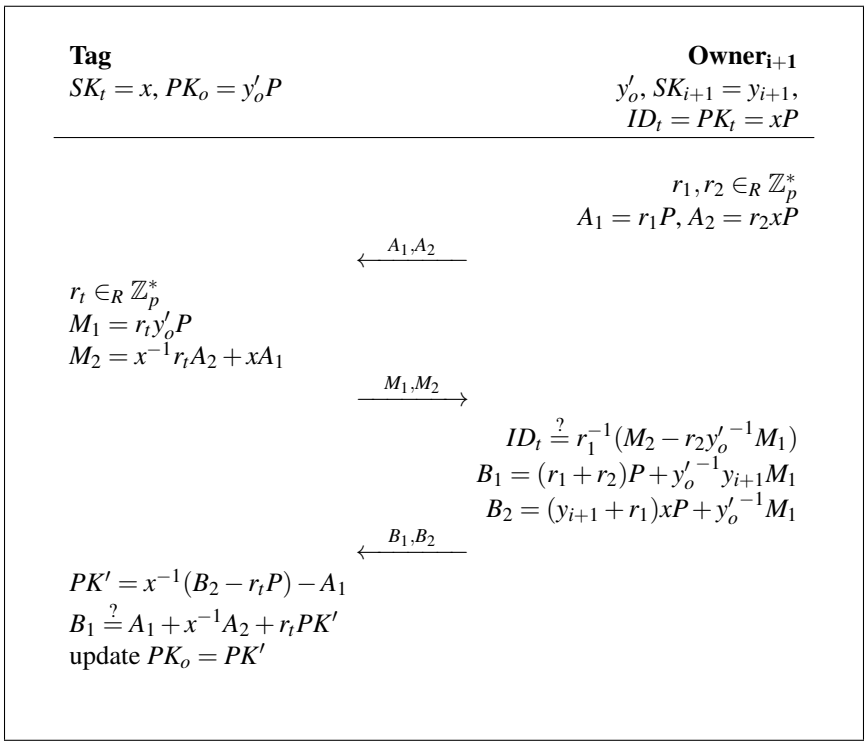

Figure 2. Key update protocol for new owner

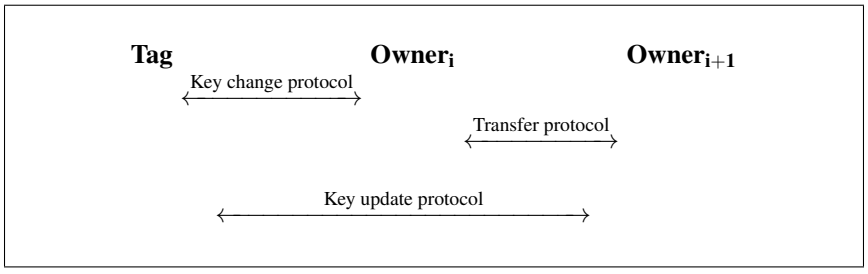

Figure 3. Protocol flows for ownership transfer

target tag, and sends the response back to the new owner. The new owner decrypts the message and get y'_o, thereby obtains the ownership of the tag.

Key Update Protocol. This protocol is executed when the new owner Owner_{i+1} obtains the ownership of the target tag. Owner_{i+1} updates the owner's public key stored in the tag with its own public key. This procotol protects the tag and the new owner from malicious previous owners so that they cannot identify or trace the interaction between the tag and the new owner after giving out the ownership. The protocol is a small modification of

key change protocol for the current owner. The protocol phase is depicted in Figure 2 and detailed as follows.

1. Owner_{i+1} randomly chooses r_1, r_2 and sends $A_1 = r_1P$, $A_2 = r_2ID_t$ to Tag.
2. Tag randomly picks a nonce r_t and responses $M_1 = r_tPK_o$, $M_2 = SK_t^{-1}r_tA_2 + SK_tA_1$.
3. Upon receiving M_1 and M_2, Owner_{i+1} computes $r_1^{-1}(M_2 - r_2y_o'^{-1}M_1)$ and checks whether the value equals ID_t. If not, Owner_{i+1} rejects Tag and terminates the protocol execution; otherwise it computes $B_1 = (r_1 + r_2)P + y_o'^{-1}SK_{i+1}M_1$ and $B_2 = (SK_{i+1} + r_1)PK_t + y_o'^{-1}M_2$, and sends them to Tag. Owner_{i+1} will keep y_o' until it succeeds in communicating with *Tag* in future interactions.
4. Tag calculates $A_1 + SK_t^{-1}A_2 + r_t(SK_t^{-1}(B_2 - r_tP) - A_1)$ and checks whether the result equals to B_1. If so, it updates PK_o with the value $SK_t^{-1}(B_2 - r_tP) - A_1$; otherwise Tag terminates the protocol execution.

The combination of key change protocol, transfer protocol and key update protocol is illustrated in Fig 3.

Controlled Delegation Protocol. Our delegation protocol uses counter stored in the tag, like [4,13], to control the delegation phases. The current owner Owner sends the maximum number of queries that can be made to Tag. After each query sent by the delegate (denoted by Delegate), Tag increases the inside counter by 1. Once the counter reaches the maximum value set by Owner, or Tag receives the cancellation command from Owner, the delegation will be terminated. The details of the controlled delegation protocol are described as follows.

1. First, Owner randomly chooses r_1, r_2 and sends $A_1 = r_1P$, $A_2 = r_2ID_t$ to Tag.
2. Then, Tag generates a nonce r_t and sends the following information: $M_1 = r_tPK_o$, $M_2 = SK_t^{-1}r_tA_2 + SK_tA_1$ to Owner.
3. Upon receiving M_1 and M_2, Owner verifies whether ID_t equals the value of $r_1^{-1}(M_2 - r_2SK_i^{-1}M_1)$. If not, Owner rejects Tag and terminates the protocol execution; otherwise it generates a private key y_d for temporary delegation, and computes $B_1 = (r_1 + r_2)P + SK_i^{-1}y_dM_1$ and $B_2 = y_dPK_t + SK_i^{-1}M_1$. Owner also picks c_m, which is the maximum number of queries that can be made to Tag and calculates $S = c_mr_2P + r_1xP$. Owner_i sends B_1, B_2, c_m and S to Tag. It also sends y_d to $Delegate$ securely to authorise the delegation.
4. Tag computes $A_1 + SK_t^{-1}A_2 + r_tSK_t^{-1}(B_2 - r_tP)$ and checks whether the value equals to B_1. It also compares S with $c_mx^{-1}A_2 + xA_1$. If both results are valid, Tag then stores the c_m and the current owner's public key PK_o, and computes the temporary delegation key $SK_t^{-1}(B_2 - r_tP)$; otherwise Tag terminates the protocol execution. The step 1 to 4 are presented in Fig 4.
5. Delegate interacts Tag with the delegation key SK_d given by Owner. Each time after being queried by Delegate, Tag adds the counter c by 1. Once c reaches the maximum value c_m, Tag replaces the delegation public key with Owner's public key so that no further queries can be made by Delegate. This procedure is described in Fig 5.

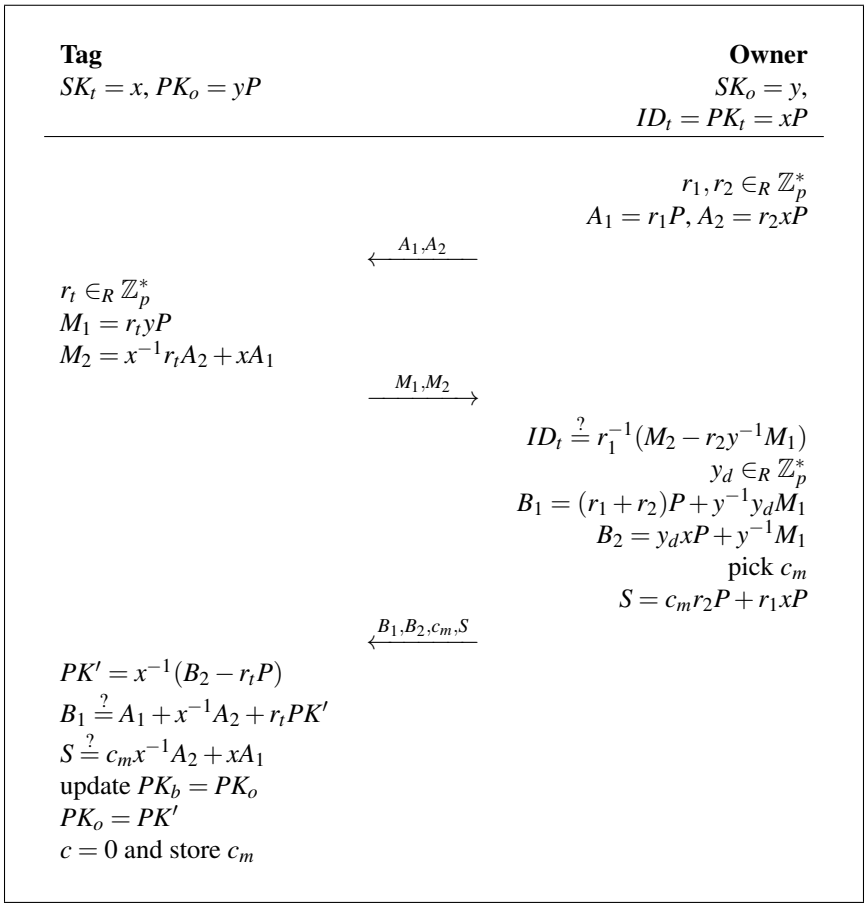

Figure 4. Controlled delegation protocol - 1

6. Owner is also able to cancel the delegation protocol at any time. It chooses a random r and sends the tuple $(r, (ry_d + y)P)$ as a cancellation request to Tag. After verifying the validity of the request, Tag replaces the delegation public key with the stored Owner's public key and cancel the delegation. This procedure is showed in Fig 6.

5. Security Analysis

5.1. Resistance to Attacks

Our scheme is secure against the attacks mentioned in 3.2.

1. Fresh nonces are used in our scheme to prevent replay attacks. An adversary is unable to gain privileges by reusing an expired message.
2. Man-in-the-middle attacks are avoided because the tag and the owner in the scheme authenticate each other using the public-private key pairs, which provides

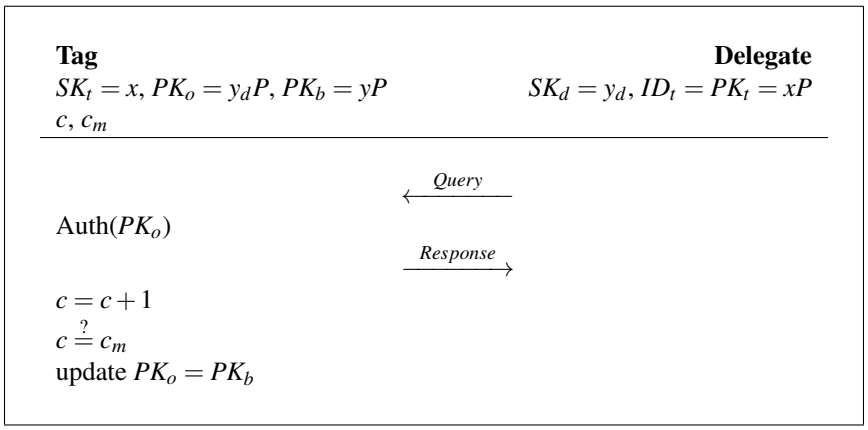

Figure 5. Controlled delegation protocol - 2

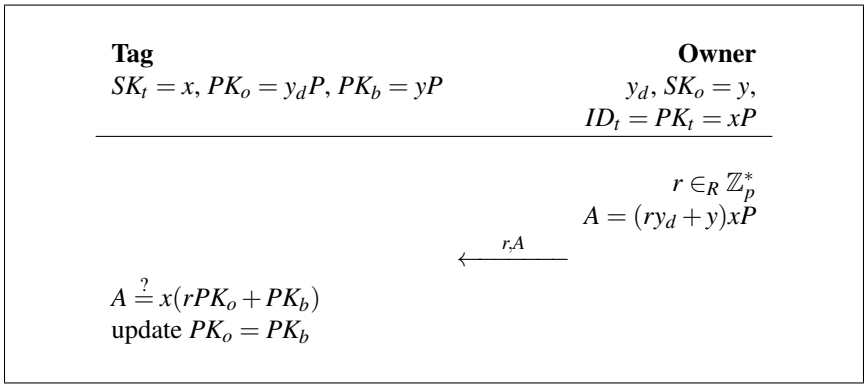

Figure 6. Controlled delegation protocol - 3

the correctness of the messages so that an adversary cannot counterfeit any message that is valid.

3. Denial-of-service attacks occur when the tag and the reader are updating the keys. An adversary can block the message sent by the owner from reaching the tag in every protocol phases in order to desynchronise the tag and the owner. However, our scheme resists against denial-of-service attacks in all the protocol phases. In key change protocol phase and the controlled delegation protocol phase, if an adversary blocks the messages and causes the tag's failure to update to the temporary public keys, the current owner is still able to communicate with the tag and generate a new ownership transfer or controlled delegation key pair for the new owner or delegate. In key update protocol phase, the new owner keeps the temporary private key until it succeeds to communicate with the tag using its own public-private key pair in future queries. In addition, the tag always verifies the messages before updating the owner's public key. Hence, an adversary is unable to manipulate the messages and cause the tag and the reader to lose synchronisation.

4. Backward traceability and forward traceability are resisted by our scheme. An adversary will not be able to decipher the past or future messages between the tag and any owner even though it knows the tag's private key. This is because all the messages involving the tag's ID (i.e. the tag's public key) are either encrypted by the owner's private key or protected by a randomly generated session nonce, which is not publicly transmitted and only known by the sender. Since finding the discrete logarithm of a elliptic curve point is infeasible, the adversary is unlikely to identify the tag from the past or future transactions.

5.2. Privacy Preservation for Owners

Previous owner privacy and new owner privacy are guaranteed by key change protocol and key update protocol.

The previous owner randomly chooses a temporary public-private key pair and changes its public key stored in the tag before transferring the ownership to the new owner. After the ownership is transferred, the new owner is not able to reveal the past transactions between the tag and the old owner because there is no link between the temporary key and the key of the previous owner. Hence previous owner privacy is effectively assured.

Later in the key update protocol, after authenticating the tag, the new owner sends a change request to update the temporary public key in the tag. Since the new owner's public key is protected by the tag's private key and a fresh nonce, it is unlikely the previous owner can extract or change the new owner's private key. Hence the future communications between the tag and the new owner is protected from the previous owner and new owner privacy is also assured.

5.3. Controlled Delegation and Authorisation Recovery

In our scheme, the present owner gives the delegation key to a third party for delegation procedure. Since the key is temporarily generated, there is no linkage between the delegation key and the owner's key. Also note that the tag stores the owner's public key. Therefore, when the queries made by the delegate reaches the allowed times, or the owner sends the command for cancelling delegation, the owner can always regain the full control over the tag. The delegate must request the owner for further access to the tag after the allowed queries are made.

Authorisation recovery is a special case of controlled delegation. The current owner stores the temporary key pair when it obtains the ownership from the previous owner. In the step 3 of controlled delgation protocol, the current owner simply transfers the temporary public key to the tag instead of a randomly chosen delegation key. It also does not need to send the private key to the previous owner since the previous owner is the one who generates the temporary key. As a result, the previous owner and the tag can communicate with each other using the temporary key pair. Just like the controlled delegation process, the authorisation recovery will expire when the counter in the tag reaches the maximum number set by the current owner, or when the tag receives the cancellation request from the current owner.

6. Performance Aspects

The proposed tag ownership transfer scheme depends on elliptic curve cryptography. The protocol phases use the operations, including modular additions and multiplications, and point multiplication on an elliptic curve. Among these three, point multiplication is the most complicated operation for passive tags. Our scheme can be easily implemented in the lightweight RFID processor architecture presented by Lee *et al.* in 2010 [9]. The RFID processor consists of a micro controller, a bus manager and a elliptic curve processor. With the operating frequency of 700 KHz, the power consumption of the processor is 13.8 μW and the cycles is $59,790$ per elliptic curve point multiplication. The performance results for our scheme are illustrated in Table 2. Therefore, our scheme is feasible for passive tags and any phases of our proposed scheme can be completed in less than 800 ms.

Table 2. Performance results of our protocols

Protocol phases	Point multiplications	Cycles	Time (ms)
Key change protocol	7	$418,530$	598
Key update protocol	7	$418,530$	598
Controlled delegation protocol - delegate	9	$538,110$	769
Controlled delegation protocol - cancellation	2	$119,580$	171

7. Conclusion

We have proposed a new RFID ownership transfer protocol in this paper. We describe the different phases of the protocol, such as key change protocol, transfer potocol, key update protocol and controlled delegation protocol. We have carried out analysis of the proposed protocol and have shown that it meets the required security and privacy features. The ownership transfer process is performed without a trusted third party. It also allows controlled delegation and authorisation recovery. Our scheme is feasible for lightweight RFID tags in terms of power consumption and processing time. To the best of our knowledge, our scheme is the first elliptic-curve based secure ownership transfer protocol.

Acknowledgements

The authors would like to thank anonymous referees for helpful comments. This work is supported by the Australian Research Council Discovery Project DP110101951.

References

[1] Lejla Batina, Jorge Guajardo, Tim Kerins, Nele Mentens, Pim Tuyls, and Ingrid Verbauwhede. Public-key cryptography for RFID-tags. In *PerCom Workshops*, pages 217–222, 2007.

[2] Kaoutar Elkhiyaoui, Erik-Oliver Blass, and Refik Molva. ROTIV: RFID ownership transfer with issuer verification. In *RFIDSec*, pages 163–182, 2011.

[3] Albert Fernàndez-Mir, Rolando Trujillo-Rasua, Jordi Castellà-Roca, and Josep Domingo-Ferrer. A scalable RFID authentication protocol supporting ownership transfer and controlled delegation. In *RFIDSec*, pages 147–162, 2011.

[4] Sepideh Fouladgar and Hossam Afifi. A simple privacy protecting scheme enabling delegation and ownership transfer for RFID tags. *Journal of Communications*, 2(6):6–13, 2007.

[5] Daniel M. Hein, Johannes Wolkerstorfer, and Norbert Felber. ECC is ready for RFID - a proof in silicon. In *Selected Areas in Cryptography*, pages 401–413, 2008.

[6] Ari Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[7] Gaurav Kapoor, Wei Zhou, and Selwyn Piramuthu. Multi-tag and multi-owner RFID ownership transfer in supply chains. *Decision Support Systems*, 52(1):258–270, 2011.

[8] Lars Kulseng, Zhen Yu, Yawen Wei, and Yong Guan. Lightweight mutual authentication and ownership transfer for RFID systems. In *INFOCOM*, pages 251–255, 2010.

[9] Yong Ki Lee, Lejla Batina, Dave Singelée, and Ingrid Verbauwhede. Low-cost untraceable authentication protocols for RFID. In *WISEC*, pages 55–64, 2010.

[10] Yong Ki Lee, Kazuo Sakiyama, Lejla Batina, and Ingrid Verbauwhede. Elliptic-curve-based security processor for RFID. *IEEE Transactions on Computers*, 57(11):1514–1527, 2008.

[11] Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In *ICICS*, pages 1–20, 2006.

[12] David Molnar, Andrea Soppera, and David Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In *Selected Areas in Cryptography*, pages 276–290, 2005.

[13] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. Practical RFID ownership transfer scheme. *Journal of Computer Security*, 19(2):319–341, 2011.

[14] Kyosuke Osaka, Tsuyoshi Takagi, Kenichi Yamazaki, and Osamu Takahashi. An efficient and secure RFID security method with ownership transfer. In *CIS*, pages 778–787, 2006.

[15] Pedro Peris-Lopez, Julio C. Hernandez-Castro, Juan E. Tapiador, Tieyan Li, and Yingjiu Li. Vulnerability analysis of RFID protocols for tag ownership transfer. *Computer Networks*, 54(9):1502–1508, 2010.

[16] Junichiro Saito, Kenji Imamoto, and Kouichi Sakurai. Reassignment scheme of an RFID tag's key for owner transfer. In *EUC Workshops*, pages 1303–1312, 2005.

[17] Boyeon Song. RFID tag ownership transfer. In *RFIDSec*, 2008.