Revisit of McCullagh–Barreto Two-Party ID-Based Authenticated Key Agreement Protocols

Kim-Kwang Raymond Choo

Information Security Research Centre
Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia
k.choo@qut.edu.au

December 2, 2004

Abstract

The recently proposed two-party ID-based authenticated key agreement protocols (with and without escrow) and its variant resistant to key-compromise impersonation by McCullagh & Barreto are revisited. The protocol carries a proof of security in the Bellare & Rogaway (1993) model. In this paper, it is demonstrated that the protocols and its variant are not secure if the adversary is allowed to send a Reveal query to reveal non-partner players who had accepted the same session key.

1 Introduction

Cryptographic protocols are the sine qua non of many diverse secure electronic commerce applications in today's globalising electronic commerce landscape. However, the design of cryptographic protocols is notoriously hard and errors were frequently found in protocols years after they were published [6, 7, 8, 9].

In a recent work, McCullagh & Barreto [5] propose a new two-party identity-based authenticated key agreement (2P-IDAKA) protocol with a proof of security in the Bellare & Rogaway (1993) model (hereafter referred to as the BR93 model) [2]. The BR93 model is one of the widely used proof models in the computational complexity approach for protocol analysis [1, 2, 4, 7]. Xie (2004) correctly pointed out a flaw in the 2P-IDAKA protocol, where a malicious adversary is able to successfully launch a key compromise attack on the protocol [10]. To address this attack pointed out by Xie, McCullagh & Barreto propose a fix resistant to key-compromise impersonation in their paper [5].

In this paper, it is demonstrated that the 2P-IDAKA protocols (with and without escrow) and the fix (variant) are not secure if the adversary is allowed to reveal non-partner players who had accepted the same session key. The remainder of this paper is structured as follows: Section 2 briefly explains the BR93 model. Section 3 describes both the 2P-IDAKA protocol and the fix, and the attack sequences on both the protocols. Section 4 presents the conclusions.

2 Overview of the BR93 Model

The BR93 model defines provable security for entity authentication and key distribution goals. The adversary $\mathcal{A}$ in the model, is a probabilistic machine that controls all the communications that take place between parties by interacting with a set of $\Pi^i_{U_1,U_2}$ oracles ($\Pi^i_{U_1,U_2}$ is defined to be the i^{th} instantiation of a principal U_1 in a specific protocol run and U_2 is the principal with whom U_1 wishes to establish a secret key). The predefined oracle queries are described informally as follows.

- The $\mathsf{Send}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i}, \mathsf{m})$ query allows $\mathcal{A}$ to send some message m of her choice to either the client $\Pi^i_{U_1,U_2}$ at will. $\Pi^i_{U_1,U_2}$, upon receiving the query, will compute what the protocol specification demands and return to $\mathcal{A}$ the response message and/or decision. If $\Pi^i_{U_1,U_2}$ has either accepted with some session key or terminated, this will be made known to $\mathcal{A}$.

- The $\mathsf{Reveal}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i})$ query allows $\mathcal{A}$ to expose an old session key that has been previously accepted. $\Pi^i_{U_1,U_2}$, upon receiving this query and if it has accepted and holds some session key, will send this session key back to $\mathcal{A}$.

- The $\mathsf{Corrupt}(\mathsf{U_1}, \mathsf{K_E})$ query allows $\mathcal{A}$ to corrupt the principal U_1 at will, and thereby learn the

complete internal state of the corrupted principal. The corrupt query also gives $\mathcal{A}$ the ability to overwrite the long-lived key of the corrupted principal with any value of her choice (i.e. K_E). This query can be used to model the real world scenarios of an insider cooperating with the adversary or an insider who has been completely compromised by the adversary.

- The $\mathsf{Test}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i})$ query is the only oracle query that does not correspond to any of $\mathcal{A}$'s abilities. If $\Pi^i_{U_1,U_2}$ has accepted with some session key and is being asked a $\mathsf{Test}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i})$ query, then depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.

The notion of freshness is used to identify the session keys about which $\mathcal{A}$ ought not to know anything because $\mathcal{A}$ has not revealed any oracles that have accepted the key and has not corrupted any principals knowing the key. Oracle $\Pi^i_{A,B}$ is fresh (or it holds a fresh session key) at the end of execution, if, and only if, oracle $\Pi^i_{A,B}$ has accepted with or without a partner oracle $\Pi^j_{B,A}$, both oracle $\Pi^i_{A,B}$ and its partner oracle $\Pi^j_{B,A}$ (if such a partner oracle exists) have not been sent a Reveal query, and the principals A and B of oracles $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ (if such a partner exists) have not been sent a $\mathsf{Corrupt}$ query.

Security is defined using the game $\mathcal{G}$, played between a malicious adversary $\mathcal{A}$ and a collection of $\Pi^i_{U_x,U_y}$ oracles for players $U_x, U_y \in \{U_1, \ldots, U_{N_p}\}$ and instances $i \in \{1, \ldots, N_s\}$. The adversary $\mathcal{A}$ runs the game simulation $\mathcal{G}$, whose setting is as follows.

- **Stage 1:** $\mathcal{A}$ is able to send any $\mathsf{SendClient}$, $\mathsf{SendServer}$, Reveal, and $\mathsf{Corrupt}$ oracle queries at will in the game simulation $\mathcal{G}$.
- **Stage 2:** At some point during $\mathcal{G}$, $\mathcal{A}$ will choose a fresh session on which to be tested and send a Test query to the fresh oracle associated with the test session. Note that the test session chosen must be fresh. Depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.
- **Stage 3:** $\mathcal{A}$ continues making any $\mathsf{SendClient}$, $\mathsf{SendServer}$, Reveal, and $\mathsf{Corrupt}$ oracle queries of its choice.
- **Stage 4:** Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

Success of $\mathcal{A}$ in $\mathcal{G}$ is measured in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ receives the real key or a random value. $\mathcal{A}$ wins if, after asking a $\mathsf{Test}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i})$ query, where $\Pi^i_{U_1,U_2}$ is fresh and has accepted, $\mathcal{A}$'s guess bit b' equals the bit b selected during the $\mathsf{Test}(\mathsf{U_1}, \mathsf{U_2}, \mathsf{i})$ query. Let the advantage function of $\mathcal{A}$ be denoted by $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$, where $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k}) = 2 \times \mathsf{Pr}[\mathsf{b} = \mathsf{b}'] - 1$.

A protocol is secure in the BR93 model if both the validity and indistinguishability requirements are satisfied:

1. Validity: When the protocol is run between two oracles in the absence of a malicious adversary, the two oracles accept the same key.
2. Indistinguishability: For all probabilistic, polynomial-time (PPT) adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$ is negligible.

3 McCullagh–Barreto Protocols

In this section, the 2P-IDAKA protocols (with and without escrow) and its variant due to McCullagh & Barreto are revisited and executions of the protocols in the presense of a malicious adversary demonstrated. Using the executions of the protocols, it is then demonstrated that the protocols are not secure if the adversary is allowed to send a Reveal oracle query.

3.1 2P-IDAKA Protocols

The 2P-IDAKA protocols and its variant due to McCullagh & Barreto are shown in Figures 1 and 2 respectively. There are two entities in the protocols, namely an initiator player A and a responder player B. The 2P-IDAKA protocols shown in Figure 1 carry a proof of security in the BR93 model.

Remark 1 *[Theorem 1 and Definition 1 in [5]] The 2P-IDAKA protocols shown in Figure 1 are secure authenticated key establishment protocols if all of the following conditions are fulfilled:*

1. *both partner oracles accept the same session key in the absence of a malicious adversary,*
2. *both uncorrupted partner oracles (i.e., having matching conversations) accept and hold the same key, and*
3. $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$ *is negligible.*

Notation used in the protocols is as follows: $(s + a)P$ denotes the public key of A, $A_{pri} = ((s + a))^{-1}P$ denotes the private key of A, $(s + b)P$ denotes the public key of B, and $B_{pri} = ((s + b))^{-1}P$ denotes the private key of B, x_a and x_b denote random nonces where $x_a, x_b \in_R Z^*_r$.

1. $\mathsf{A} \longrightarrow \mathsf{B}:$	$A_{KA} = x_a(s+b)P$
2. $\mathsf{B} \longrightarrow \mathsf{A}:$	$B_{KA} = x_b(s+a)P$
A computes session key $e(B_{KA}, A_{pri})^{x_a} = e(P,P)^{x_a x_b}$.	
B computes session key $e(A_{KA}, B_{pri})^{x_b} = e(P,P)^{x_a x_b}$.	

Figure 1: McCullagh–Barreto 2P-ID-AKA protocols with and without escrow

1. $\mathsf{A} \longrightarrow \mathsf{B}:$	$A_{KA} = x_a(s+b)P$
2. $\mathsf{B} \longrightarrow \mathsf{A}:$	$B_{KA} = x_b(s+a)P$
A computes session key $e(P,P)^{x_a} e(B_{KA}, A_{pri}) = e(P,P)^{x_a+x_b}$.	
B computes session key $e(P,P)^{x_b} e(A_{KA}, B_{pri}) = e(P,P)^{x_a+x_b}$.	

Figure 2: Proposed fix to Xie (2004)'s attack

3.2 Attacks on the Protocols

As Blake–Wilson, Johnson, & Menezes (1997) [3] had correctly pointed out, two-flow authenticated key establishment protocols that do not contain asymmetry in the formation of the session key will not meet the security requirements in the BR93 model. Figures 3 and 4 illustrate an execution of the protocols in the presence of a malicious adversary $\mathcal{A}$.

1. $\mathsf{A} \longrightarrow \mathsf{B}:$	$x_a(s+b)P$
The adversary $\mathcal{A}$ intercepts and deletes the message $A_{KA} = x_a(s+b)P$. $\mathcal{A}$ then chooses a random nonce $x_E \in_E Z_r^*$ and sends message $x_a(s+b)P \cdot x_E$ impersonating A.	
1. $\mathcal{A}_{\mathsf{A}} \longrightarrow \mathsf{B}:$	$x_a(s+b)P \cdot x_E$
2. $\mathsf{B} \longrightarrow \mathsf{A}:$	$x_b(s+a)P$
The adversary $\mathcal{A}$ intercepts and deletes the message $B_{KA} = x_b(s+a)P$.	
2. $\mathcal{A}_{\mathsf{B}} \longrightarrow \mathsf{A}:$	$x_b(s+a)P \cdot x_E$
A computes session key $e(x_b(s+a)P \cdot x_E, A_{pri})^{x_a} = e(P,P)^{x_a x_b x_E}$.	
B computes session key $e(x_a(s+b)P \cdot x_E, B_{pri})^{x_b} = e(P,P)^{x_a x_b x_E}$.	
$\mathcal{A} \longrightarrow \mathsf{B}:$	Reveal
$\mathsf{B} \longrightarrow \mathcal{A}:$	$e(P,P)^{x_a x_b x_E}$

Figure 3: Attack sequence on the McCullagh–Barreto 2P-ID-AKA protocols

1. $\mathsf{A} \longrightarrow \mathsf{B}:$	$A_{KA} = x_a(s+b)P$
The adversary $\mathcal{A}$ intercepts and deletes the message $A_{KA} = x_a(s+b)P$.	
1. $\mathcal{A}_{\mathsf{A}} \longrightarrow \mathsf{B}:$	$A_{KE} = x_a(s+b)P + x_E(s+b)P$
2. $\mathsf{B} \longrightarrow \mathsf{A}:$	$B_{KA} = x_b(s+a)P$
The adversary $\mathcal{A}$ intercepts and deletes the message $B_{KA} = x_b(s+a)P$.	
2. $\mathcal{A}_{\mathsf{B}} \longrightarrow \mathsf{A}:$	$B_{KE} = x_b(s+a)P + x_E(s+a)P$
A computes session key $e(P,P)^{x_a} e(B_{KE}, A_{pri}) = e(P,P)^{x_a+x_b+x_E}$.	
B computes session key $e(P,P)^{x_b} e(A_{KE}, B_{pri}) = e(P,P)^{x_a+x_b+x_E}$.	
$\mathcal{A} \longrightarrow \mathsf{B}:$	Reveal
$\mathsf{B} \longrightarrow \mathcal{A}:$	$e(P,P)^{x_a+x_b+x_E}$

Figure 4: Attack sequence on the proposed fix

In the attack sequences shown in Figures 3 and 4, both A and B have accepted the same session key. However, both A and B are non-partners since they do not have matching conversations. In addition, both A and B are uncorrupted since they have not been sent a Corrupt query[1].

By sending a Reveal query to either A or B, the malicious adversary $\mathcal{A}$ is able to obtain the session key of a fresh oracle of a non-partner oracle, as shown in Figures 3 and 4.

Remark 2 *Hence, the 2P-IDAKA protocols and its variant shown in Figures 1 and 2 are not secure since the malicious adversary $\mathcal{A}$ is able to obtain the session key of a fresh oracle of a non-partner oracle by revealing a non-partner oracle holding the same key (i.e., violating the key establishment goal).*

4 Conclusion

Through a detailed study of the McCullagh–Barreto 2P-IDAKA protocols and its variant, it is demonstrated that the protocols and its variant are insecure if the adversary is allowed to reveal non-partner players who share the same session key and obtain a fresh session key. Trivially this implies the violation of the key establishment goal.

References

[1] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated Key Exchange Secure

[1] Note that an oracle is considered opened but not corrupted if it has only been sent a Reveal query.

Against Dictionary Attacks. In *Advances in Cryptology – Eurocrypt 2000*, pages 139 – 155. Springer-Verlag, 2000. Volume 1807 of Lecture Notes in Computer Science.

[2] Mihir Bellare and Phillip Rogaway. Entity Authentication and Key Distribution. In *Advances in Cryptology - Crypto 1993*, pages 110–125. Springer-Verlag, 1993. Volume 773 of Lecture Notes in Computer Science.

[3] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key Agreement Protocols and their Security Analysis. In *6th IMA International Conference on Cryptography and Coding*, pages 30–45. Springer-Verlag, 1997. Volume 1355 of Lecture Notes in Computer Science.

[4] Ran Canetti and Hugo Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Advances in Cryptology - Eurocrypt 2001*, pages 453–474. Springer-Verlag, May 2001. Volume 2045 of Lecture Notes in Computer Science.

[5] Noel McCullagh and Paulo S. L. M. Barreto. A New Two-Party Identity-Based Authenticated Key Agreement. In *(To Appear in) Cryptographers' Track at RSA Conference - CT-RSA 2005. Available from http://eprint.iacr.org/2004/122/.* Springer-Verlag, 2005. Lecture Notes in Computer Science.

[6] Junghyun Nam, Seungjoo Kim, and Dongho Won. Attacks on Bresson-Chevassut-Essiari-Pointcheval's Group Key Agreement Scheme. Cryptology ePrint Archive, Report 2004/251, 2004. http://eprint.iacr.org/2004/251/.

[7] Victor Shoup. OAEP Reconsidered. In *Advances in Cryptology - Crypto 2001*, pages 239–259. Springer-Verlag, 2001. Volume 2139 of Lecture Notes in Computer Science.

[8] Zhiguo Wan and Shuhong Wang. Cryptanalysis of Two Password-Authenticated Key Exchange Protocols. In *9th Australasian Conference on Information Security and Privacy - ACISP 2004*. Springer-Verlag, 2004. Volume 3108 of Lecture Notes in Computer Science.

[9] Duncan S. Wong and Agnes H. Chan. Efficient and Mutually Authenticated Key Exchange for Low Power Computing Devices. In *Advances in Cryptology - Asiacrypt 2001*, pages 172–289. Springer-Verlag, 2001. Volume 2248 of Lecture Notes in Computer Science.

[10] Guohong Xie. Cryptanalysis of Noel McCullagh and Paulo S. L. M. Barreto Two-Party Identity-Based Key Agreement. Cryptology ePrint Archive, Report 2004/308, 2004. http://eprint.iacr.org/2004/308/.