

# On the Security of an Improved Password Authentication Scheme Based on ECC

Ding Wang<sup>1,2\*</sup>, Chun-guang Ma<sup>1\*\*</sup>, and Yu-heng Wang<sup>2,3</sup>

<sup>1</sup> Harbin Engineering University, Harbin City 150001, China

<sup>2</sup> Automobile Management Institute of PLA, Bengbu City 233011, China

<sup>3</sup> Golisano College of Computing and Information Sciences, Rochester Institute of Technology, 102 Lomb Memorial Dr., Rochester, NY 14623, USA  
wangdingg@mail.nankai.edu.cn

**Abstract.** The design of secure remote user authentication schemes for mobile applications is still an open and quite challenging problem, though many schemes have been published lately. Recently, Islam and Biswas pointed out that Lin and Hwang et al.'s password-based authentication scheme is vulnerable to various attacks, and then presented an improved scheme based on elliptic curve cryptography (ECC) to overcome the drawbacks. Based on heuristic security analysis, Islam and Biswas claimed that their scheme is secure and can withstand all related attacks. In this paper, however, we show that Islam and Biswas's scheme cannot achieve the claimed security goals and report its flaws: (1) It is vulnerable to offline password guessing attack, stolen verifier attack and denial of service (DoS) attack; (2) It fails to preserve user anonymity. The cryptanalysis demonstrates that the scheme under study is unfit for practical use.

**Keywords:** Authentication protocol; Smart card; Cryptanalysis; User anonymity; Elliptic curve cryptography

## 1 Introduction

With the large-scale proliferation of internet and network technologies over the last couple of decades, more and more electronic transactions for mobile devices are implemented on Internet or wireless networks. It is increasingly important to protect the systems and the users' privacy and security from malicious adversaries. Accordingly, user authentication becomes an essential security mechanism for remote systems to assure one communicating party of the legitimacy of the corresponding party by acquisition of corroborative evidence. Authentication factors are generally grouped into three categories [1]: 1) what you know (e.g., passwords, PINs); 2) what you have (e.g., tokens, smart card); and 3) who you are (e.g., biometric). Among numerous methods based on one or more of these

---

\* Corresponding author.

\*\* The abridged version of this paper is going to appear in the proceedings of ICICA 2012, LNCS, Springer-Verlag.

three types, password authentication is one of the most simple and effective approaches for authentication in a client/server environment, and has been widely deployed in areas like e-banking, e-commerce, e-health services and so on.

Since Lamport [2] introduced the first password-based authentication scheme in 1981, many password-based remote user authentication schemes [3–6] have been proposed, where a client remembers a password and the corresponding server holds the password or its verification data that are used to verify the client’s knowledge of the password. These easy-to-remember passwords, called weak passwords, have low entropy and thus are potentially vulnerable to various sophisticated attacks, especially offline password guessing attack [7], which is the gravest threat a well-designed password authentication scheme must be able to thwart. A common feature among the published schemes is that computation efficiency and system security cannot be achieved at the same time. As the computation ability and battery capacity of mobile devices (e.g. PDAs, smart cards) are limited, the traditional public-key based remote authentication schemes are not suitable for mobile applications.

Fortunately, it seems to see the dawn in recent two years, where several schemes based on ECC have been proposed to reduce computation cost while preserving security strength [8–12]. However, The reality of the situation is that this dilemma is only partially addressed and most of the ECC-based schemes were found severely flawed shortly after they were first put forward, so intensive further research is required. More recently, Islam and Biswas [13] proposed an advanced password authentication scheme based on ECC. The authors claimed that their scheme provides mutual authentication and is free from all known cryptographic attacks, such as replay attack, offline password guessing attack, insider attack and so on. Although their scheme is superior to the previous solutions for implementation on mobile devices, we find their scheme cannot achieve the claimed security: their scheme is vulnerable to the offline password guessing attack, the stolen verifier attack. Almost at the same time with us, He et al. [14] also have identified these defects in Islam-Biswas’s scheme. Hence, we went on to perform a further cryptanalysis on this protocol and observe that it is also prone to a denial of service (DoS) attack, and it transmits user’s identity in plain during the login request and thus user anonymity is not provided, while provision of user identity confidentiality is of great importance for a protocol in mobile environments [15].

The remainder of this paper is organized as follows: in Section 2, we review Islam-Biswas’s scheme. Section 3 describes the weaknesses of Islam-Biswas’s scheme. Section 4 concludes the paper.

## 2 Review of Islam-Biswas’s scheme

In this section, we examine the password authentication scheme using smart cards proposed by Islam and Biswas [13] in 2011. Islam-Biswas’s scheme, summarized in Fig.1, consists of four phases: the registration phase, the authentication phase, the session key distribution phase and the password change phase.

For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 1.

**Table 1.** Notations

Symbol	Description
$U_i$	$i^{th}$ user
$S$	remote server
$ID_i$	identity of user $U_i$
$PW_i$	password of user $U_i$
$d_s$	secret key of remote server $S$
$O$	the point at infinity
$G$	base point of the elliptic curve group of order $n$ such that $n \cdot G = O$
$V_s$	public key of remote server $S$ , where $V_s = d_s \cdot G$
$V_i$	password-verifier of $U_i$ , where $V_i = PW_i \cdot G$
$K_x$	secret key computed using $K = d_s \cdot V_i = PW_i \cdot V_s = (K_x, K_y)$
$E_{K_x}(\cdot)$	symmetric encryption with $K_x$
$\oplus$	the bitwise XOR operation
$\parallel$	the string concatenation operation
$H(\cdot)$	collision free one-way hash function
$A \rightarrow B : M$	message $M$ is transferred through a common channel from $A$ to $B$
$A \Rightarrow B : M$	message $M$ is transferred through a secure channel from $A$ to $B$

## 2.1 Registration phase

Before the system begins, the server  $S$  selects a large prime number  $p$  and two integer elements  $a$  and  $b$ , where  $p > 2^{160}$  and  $4a^3 + 27b^2 \pmod{p} \neq 0$ . Then  $S$  selects an elliptic curve equation  $E_p$  over finite field  $F_p$ :  $y^2 = x^3 + ax + b \pmod{p}$ . Let  $G$  be a base point of the elliptic curve with a prime order  $n$  and  $O$  be a point at infinite, where  $n \cdot G = O$  and  $n > 2^{160}$ . The server chooses the private key  $d_s$  and computes the corresponding public key  $V_s = d_s \cdot G$ . The registration phase involves the following operations:

- 1)  $U_i$  chooses his identity  $ID_i$  and password  $PW_i$ , computes  $V_i = PW_i \cdot G$ .
- 2)  $U_i \Rightarrow S : \{ID_i, V_i\}$ .
- 3) On receiving the registration message from  $U_i$ , the server  $S$  create an entry  $(ID_i, V_i, status-bit)$  in its database, where the *status-bit* indicates the status of the client, i.e., when the client is logged-in to the server the *status-bit* is set to one, otherwise it is set to zero.

## 2.2 Login and authentication phase

When  $U_i$  wants to login to  $S$ , the following operations will be performed:

- 1)  $U_i$  keys his identity  $ID_i$  and the password  $PW_i$  into the terminal. The client selects a random number  $r_i$  from  $[1, n-1]$ , computes  $R_i = r_i \cdot V_s$  and  $W_i = (r_i \cdot PW_i) \cdot G$ . Then encrypts  $(ID_i, R_i, W_i)$  using a symmetric key  $K_x$ , where  $K_x$  is the  $x$  coordinate of  $K = PW_i \cdot V_s = (K_x, K_y)$ .
- 2)  $U_i \Rightarrow S : \{ID_i, E_{K_x}(ID_i \parallel R_i \parallel W_i)\}$
- 3)  $S$  computes the decryption key  $K_x$  by calculating  $K = d_s \cdot V_i = (K_x, K_y)$  and then decrypts  $E_{K_x}(ID_i \parallel R_i \parallel W_i)$  using  $K_x$ . Subsequently  $S$  compares decrypted  $ID_i$  with received  $ID_i$ ,  $\hat{e}(R_i, V_i)$  with  $\hat{e}(W_i, V_s)$ , respectively. If both conditions are satisfied,  $S$  selects a random number  $r_s$  and computes  $W_s = r_s \cdot V_s = r_s \cdot d_s \cdot G$ .
- 4)  $S \rightarrow U_i : \{W_i + W_s, H(W_s)\}$ .
- 5)  $U_i$  retrieves  $W_s$  by subtracting  $W_i$  from  $W_i + W_s$ . If the hashed result of retrieved  $W_s$  is equal to the received  $H(W_s)$ , then  $U_i$  performs the hash operation  $H(W_i \parallel W_s)$  and sends it to the server.
- 6)  $U_i \rightarrow S : \{H(W_i \parallel W_s)\}$ .
- 7) The server  $S$  computes the hash value with its own copies of  $W_s$  and  $W_i$  and compares it with the received  $H(W_i \parallel W_s)$ , to accept or denied the login request. If the equality holds, the server grants the client's login request, otherwise rejects.

### 2.3 Session key distribution phase and password change phase

Since both the session key distribution phase and password change phase have little relevance with our discussions, they are omitted here.

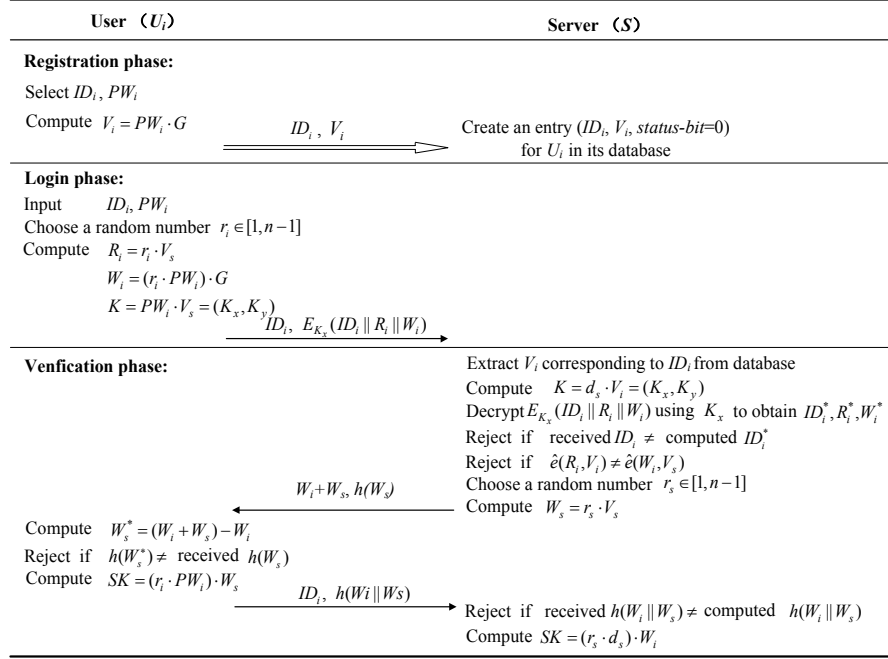
## 3 Cryptanalysis of Islam-Biswas's scheme

With superior performance over other related schemes and a long list of arguments of security features that their scheme possesses presented, Islam-Biswas's scheme seems desirable at first glance. However, their security arguments are still specific-attack-scenario-based and without some degree of rigorousness, and thus it is not fully convincing. We find that Islam-Biswas's scheme still fails to serve its purposes and demonstrate its security flaws in the following.

### 3.1 Offline password guessing attack

To successfully launch an offline password guessing attack on a remote user authentication scheme, the following two conditions must be satisfied [16]: (1) the user's password is a weak password, and (2) a piece of password-related information that can be obtained and used as a comparison target for password guessing.

In Islam-Biswas's scheme, a user is allowed to choose her own password at will during the registration and password change phases; the user usually tends to select a password, *e.g.*, his birthday or home phone number, which is easily remembered for his convenience. Hence, these easy-to-remember passwords, called



**Fig. 1.** Islam-Biswas's remote user authentication scheme

weak passwords, have low entropy and thus are potentially vulnerable to offline password guessing attack.

Besides, user  $U_i$ 's identity is transmitted in plaintext within the login request, it is not difficult for an adversary  $\mathcal{A}$  to identify the login request message sent by  $U_i$ . Once the login request message  $ID_i, E_{K_x}(ID_i \| R_i \| W_i)$  during any authentication process is intercepted by  $\mathcal{A}$ , an offline password guessing attack can be launched as follows:

- Step 1.** Guesses the value of  $PW_i$  to be  $PW_i^*$  from a dictionary space  $\mathcal{D}$ .
- Step 2.** Computes  $K^* = PW_i^* \cdot V_s = (K_x^*, K_y^*)$ , as  $V_s$  is the public key of server  $S$ .
- Step 3.** Decrypts the previously intercepted  $E_{K_x}(ID_i \| R_i \| W_i)$  using  $K_x^*$  to obtain  $ID_i^*$ .
- Step 4.** Verifies the correctness of  $PW_i^*$  by checking if the computed  $ID_i^*$  is equal to the intercepted  $ID_i$ .
- Step 5.** Repeats Step 1, 2, 3, and 4 of this procedure until the correct value of  $PW_i$  is found.

As the size of the password dictionary, i.e.  $|\mathcal{D}|$ , is very limited in practice, the above attack procedure can be completed in polynomial time. Moreover, the above attack we describe is very effective because it only requires the abilities of an eavesdropping attacker, and involves no expensive cryptographic operations.

After guessing the correct value of  $PW_i$ ,  $\mathcal{A}$  can compute the valid symmetric key  $K = PW_i \cdot V_s = (K_x, K_y)$ . Then the attacker can impersonate  $U_i$  to send a valid login request message  $\{ID_i, E_{K_x}(ID_i \parallel R_i \parallel W_i)\}$  to the service provider server  $S$ , since  $U_i$ 's identity  $ID_i$  can be intercepted from the channel,  $R_i$  and  $W_i$  can be correctly fabricated with the correctly guessed  $PW_i$ . Upon receiving the fabricated login request,  $S$  will find no abnormality and responds with  $\{W_i + W_s, H(W_s)\}$ . Then  $\mathcal{A}$  can compute the valid  $W_s$  since she knows  $W_i$ . Hence the attacker  $\mathcal{A}$  can successfully masquerade as a legitimate user  $U_i$  to server  $S$ . On the other hand, the attacker may also impersonate the server  $S$  to  $U_i$  successfully in a similar way.

### 3.2 Stolen verifier attack

Let us consider the following scenarios. In case the verifier table in the database of the server  $S$  is leaked out or stolen by an adversary  $\mathcal{A}$ . With the obtained entry  $(ID_i, V_i, status - bit)$  corresponding to  $U_i$ , she can guess out the password  $PW_i$  of  $U_i$  using the method as follows:

- Step 1.** Guesses the value of  $PW_i$  to be  $PW_i^*$  from a dictionary space  $\mathcal{D}$ .
- Step 2.** Computes  $V_i^* = PW_i^* \cdot G$ , as  $G$  is public.
- Step 3.** Verifies the correctness of  $PW_i^*$  by checking if the computed  $V_i^*$  is equal to the somehow obtained  $V_i$ .
- Step 4.** Repeats Steps 1, 2, and 3 of this procedure until the correct value of  $PW_i$  is found.

As the size of the password dictionary, i.e.  $|\mathcal{D}|$ , is very limited in practice, the above attack procedure can be completed in polynomial time. Since the underlying assumption of the above attack introduced is much constrained, it is much less effective than the attack introduced in Section 3.1. However, it is still an insecure factor to be noticed.

### 3.3 Failure of protecting the user's anonymity

As violation concern of user privacy on e-commerce and industrial engineering applications is promptly raised among individuals, human right organizations and national governments, identity protection has become a very popular research topic in recent years. Many systems have been advanced, which implement different (and sometimes even contradictory) notions of what it means to be "anonymous. Instead of a single anonymity feature, there are quite different flavors of anonymity, such as sender un-traceability, blender anonymity, sender k-anonymity and so on [17]. As for remote authentication schemes, user anonymity basically means initiator anonymity (i.e. sender anonymity), in other words, it means the adversary could not have any knowledge of real identity of the initiator but may know whether two conversations are coming from the same (unknown) participant. Comparatively, a more admired anonymity feature is initiator un-traceability (i.e., sender un-traceability), which indicates that the

attacker can neither know who the initiator is nor distinguish whether two conversations come from the same (unknown) initiator [18]. A protocol with user anonymity prevents an adversary from acquiring sensitive personal information about an individual’s preferences, lifestyles, shopping patterns, social circle, current location, etc. by analyzing the login information.

In Islam-Biswas’s scheme, the user’s identity  $ID$  is transmitted in plain, which may leak the identity of the logging user once the login messages were eavesdropped; the user’s identity  $ID$  is static in all the login phases, which may facilitate the attacker to trace out the different login request messages belonging to the same user and to derive some information related to the user  $U_i$ . In a word, neither initiator anonymity nor initiator un-traceability can be preserved in their scheme.

### 3.4 Denial of service attack

Without any knowledge of the user private information like password or security parameters stored in smart card, an adversary  $\mathcal{A}$  can successfully launch a kind of denial of service attack, which is the so called “clogging attack” [19], in many non-DoS-resilient cryptography protocols. Let’s see how this could happen with Islam-Biswas’s scheme in place. The following is performed by the adversary  $\mathcal{A}$ :

- Step 1.** Sends the previously intercepted  $\{ID_i, E_{K_x}(ID_i \parallel R_i \parallel W_i)\}$  to the server  $S$ .
- Step 2.** Ignores the reply from the server  $S$ .

The following is performed by the server:

- Step 1’.** On receiving the login request from  $U_i$  (actually  $\mathcal{A}$ ),  $S$  computes the decryption key  $K_x$  by calculating  $K = d_s \cdot V_i = (K_x, K_y)$  and then decrypts  $E_{K_x}(ID_i \parallel R_i \parallel W_i)$  using  $K_x$ . Subsequently  $S$  compares decrypted  $ID_i$  with received  $ID_i$ ,  $\hat{e}(R_i, V_i)$  with  $\hat{e}(W_i, V_s)$ , respectively.
- Step 2’.** Selects a random number  $r_s$  and computes  $W_s = r_s \cdot V_s = r_s \cdot d_s \cdot G$ .
- Step 3’.** Sends out  $\{W_i + W_s, H(W_s)\}$  and waits for the response from  $U_i$  (actually  $\mathcal{A}$ ), which will never come.

Since  $ID_i$  and  $E_{K_x}(ID_i \parallel R_i \parallel W_i)$  are valid,  $S$  will find no abnormality in Step 1’, and then proceeds to Step 2’.

The point here is that, in the above attack, the adversary  $\mathcal{A}$  doesnot need to perform any special or expensive cryptographic operations but sending one message out. However, on the server side, in Step 1’,  $S$  needs to perform one symmetric-key decryption and one bilinear pairing operation, which are computationally intensive. According to [20], the cost of one bilinear pairing operation is twenty times higher than that of one scale multiplication, and two times higher than that of one modulo exponentiation at the same security level.

It should be noted that even DoS-resilient mechanisms (e.g. timeout or locking user account for a period of time after a predefined number of login failures) are introduced on server side, it may be not a real obstacle for attacker  $\mathcal{A}$  as it can initialize new sessions with different intercepted identities in an interleaving manner. Hence,  $\mathcal{A}$  can potentially performs the above attack procedure continuously, which will make the victimized server keeps computing the useless expensive operations rather than any real work. Thus  $\mathcal{A}$  clogs  $S$  with useless work and therefore  $S$  denies any legitimate user any service. If distributed DoS attacks are launched based on this strategy, the consequences will be more serious.

## 4 Conclusion

Understanding security failures of cryptographic protocols is the key to both patching existing protocols and designing future protocols. Smartcard-based password authentication technology has been widely deployed in various kinds of security-critical applications, and careful security considerations should be taken when designing such schemes. In this paper, we have shown that Islam-Biswas's scheme suffers from the offline password guessing attack, stolen-verifier attack and denial of service attack. In addition, their scheme fails to provide the property of user anonymity. In conclusion, Although Islam-Biswas's scheme is very efficient and possesses many attractive features, it, in fact, does not provide all of the security properties that they claimed and only radical revisions of the protocol can possibly eliminate the identified flaws. Therefore, the scheme under study is not recommended for practical applications. In future work, we will propose an improvement over Islam-Biswas's scheme to overcome the identified drawbacks.

## Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions. This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042.

## References

1. O'Gorman, L.: Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE* 91(12), 2021–2040 (2003)
2. Lamport, L.: Password authentication with insecure communication. *Communications of the ACM* 24(11), 770–772 (1981)
3. Liao, I.E., Lee, C.C., Hwang, M.S.: A password authentication scheme over insecure networks. *Journal of Computer and System Sciences* 72(4), 727–740 (2006)
4. Song, R.: Advanced smart card based password authentication protocol. *Computer Standards & Interfaces* 32(5), 321–325 (2010)



5. Yeh, K.H., Su, C., Lo, N.W., Li, Y., Hung, Y.X.: Two robust remote user authentication protocols using smart cards. *Journal of Systems and Software* 83(12), 2556–2565 (2010)
6. Ma, C.G., Wang, D., Zhang, Q.M.: Cryptanalysis and improvement of Sood et al.s dynamic id-based authentication scheme. In: *Distributed Computing and Internet Technology, LNCS*, vol. 7154, pp. 141–152. Springer-Verlag (2012)
7. Klein, D.V.: Foiling the cracker: A survey of, and improvements to, password security. In: *Proceedings of the 2nd USENIX Security Workshop*. pp. 5–14 (1990)
8. Wang, R.C., Juang, W.S., Lei, C.L.: Robust authentication and key agreement scheme preserving the privacy of secret key. *Computer Communications* 34(3), 274–280 (2011)
9. Wu, S.H., Zhu, Y.F., Pu, Q.: Robust smart-cards-based user authentication scheme with user anonymity. *Security and Communication Networks* 5(2), 236–248 (2012)
10. Wei, J., Hu, X., and Liu, W.: An improved authentication scheme for telecare medicine information systems. *Journal of Medical Systems*, 1–8, 10.1007/s10916-012-9835-1 (2012)
11. Pu, Q., Wang, J., Zhao, R.: Strong authentication scheme for telecare medicine information systems. *Journal of Medical Systems* pp. 1–11 (2011), doi:10.1007/s10916-011-9735-9
12. He, D.B., Chen, J.H., Zhang, R.: A more secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems* pp. 1–7 (2012), doi:10.1007/s10916-011-9658-5
13. Islam, S.H., Biswas, G.P.: Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modelling* (2011), doi:10.1016/j.mcm.2011.07.001
14. He, D.B.: Comments on a password authentication and update scheme based on elliptic curve cryptography. *Cryptology ePrint Archive, Report 2011/411* (2011), <http://eprint.iacr.org/2011/411.pdf>
15. Wan, Z., Zhu, B., Deng, R.H., Bao, F., Ananda, A.L.: Dos-resistant access control protocol with identity confidentiality for wireless networks. In: *2005 IEEE Wireless Communications and Networking Conference (WCNC)*. vol. 3, pp. 1521–1526. IEEE (2005)
16. Horng, W.B. and Lee, C.P. and Peng, J.W.: Cryptanalysis of a more secure remote user authentication scheme. In: *2010 International Computer Symposium (ICS)*. pp. 284–287. IEEE (2010)
17. Hughes, D., Shmatikov, V.: Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security* 12(1), 3–36 (2004)
18. Li X., Qiu W., Zheng D., Chen K., and Li J.: Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Transactions on Industrial Electronics* 57(2), 793–800 (2010)
19. Roy, S., Das, A.K., Li, Y.: Cryptanalysis and security enhancement of an advanced authentication scheme using smart cards, and a key agreement scheme for two-party communication. In: *2011 IEEE 30th International Performance Computing and Communications Conference (IPCCC)*. pp. 1–7. IEEE (2011)
20. Cao, X., Kou, W., Du, X.: A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Information Sciences* 180(15), 2895–2903 (2010)