Cryptanalysis of Two PAKE Protocols for Body Area Networks and Smart Environments

Mohsen Toorani
Department of Informatics, University of Bergen
P.O. Box 7803, N-5020 Bergen, Norway
(Email: mohsen.toorani@uib.no)

(Received Nov. 23, 2014; revised and accepted May 20 & May 26, 2015)

Abstract

Password-authenticated key exchange (PAKE) protocols enable two or more entities to authenticate each other and share a strong cryptographic key based on a pre-shared human memorable password. In this paper, we present several attacks on two recent elliptic curve-based PAKE protocols that have been suggested for use in body area networks and smart environments. A variant of the first PAKE protocol has been included in the latest standard for body area networks. The second PAKE protocol is a modified variant of the first protocol, and has been proposed for bridging the user interface gap in pervasive computing and smart environments.

Keywords: Dictionary attack, elliptic curves, forward secrecy, impersonation attack, invalid-curve attack

1 Introduction

Authenticated key exchange (AKE) protocols aim to establish a cryptographic session key between legitimate entities in an authenticated manner. Many AKE protocols have been proposed in literature, but some of them have security problems [3, 17, 18]. Password-authenticated key exchange (PAKE) protocols are password-based AKE protocols that use pre-shared human memorable passwords for authentication and establishing a cryptographically strong secret key. Since introduction of the first PAKE protocol in 1992 [2], many PAKE protocols have been proposed. Many of those protocols have been shown to be insecure [7, 11, 12, 16].

Traditionally, PAKE protocols use just a shared password between a client and server. Even though people are always recommended to select strong passwords, many people choose simple passwords. As a countermeasure, many PAKE protocols try to provide multi-factor authentication by combining passwords with other parameters like public keys or symmetric keys.

Several security models have been developed for AKE and PAKE protocols, each of them has a different assumption for capabilities of an adversary. A protocol that is proved to be secure in a security model would be insecure in other security models. It is because of different assumptions on adversarial power and valid attacks.

Several security attributes should be provided by PAKE protocols, and they should withstand well-known attacks. Security requirements for PAKE protocols depend on number of participants and secret parameters that are used for constructing the protocol. Some security requirements of PAKE protocols are common with AKE protocols. This includes mutual authentication, known-key security, forward secrecy, key control, and resilience to impersonation, replay, unknown key-share (UKS), and Denning-Sacco attacks [9, 10, 16]. Furthermore, any PAKE protocol must be resilient to dictionary [5, 16] or password guessing [4] attacks. Such requirement is very subtle because people usually select weak memorable passwords. Then, instead of a brute-force attack, an attacker would use a dictionary of most probable passwords. Based on secret parameters that are used for building a protocol, there are some more requirements that should be satisfied. Those PAKE protocols that use public keys are expected to provide resilience to the key compromise impersonation (KCI) attack and its variants.

The wireless body area network (WBAN) is a wireless network of wearable computing devices [13]. WBAN

has many applications in military, ubiquitous health care, sport, and entertainment. As WBANs are resource-constrained in terms of power, memory, communication rate and computational capability, security solutions proposed for other networks may not be suitable for WBANs. The latest standardization of WBANs is the IEEE 802.15.6 standard [1], but it has security problems [19].

Ho [8] presented four elliptic curve-based key agreement protocols that are designed for different device configurations in WBAN, and can be implemented as a versatile suite through a single implementation. It includes one unauthenticated key exchange protocol, one AKE protocol with out-of-band transfer of public key, one PAKE protocol, and one AKE protocol for devices with numerical display. Variants of those protocols have been included in the IEEE 802.15.6 standard. However, there are two major differences between Ho's protocols and the protocols in the IEEE 802.15.6 standard. The first difference is that Ho's protocols do not consider validation of public keys which makes the protocols vulnerable to some extra attacks, while the protocols in the standard have considered public key validations. The second difference is in sending a masked public key in the corresponding PAKE protocols.

It has been shown [19] that the key agreement protocols in the IEEE 802.15.6 standard are vulnerable to some attacks: The unauthenticated key exchange protocol (Protocol I) is vulnerable to an impersonation attack; the AKE protocol with hidden public key transfer (Protocol II) is vulnerable to a KCI attack; the PAKE protocol (Protocol III) is vulnerable to an impersonation attack and an offline dictionary attack; and the AKE protocol for devices with numerical display (Protocol IV) is vulnerable to an impersonation attack.

All the attacks on Protocols I, II, and IV are applicable to the corresponding protocols in [8], because the protocols are almost the same. However, Ho's PAKE protocol has different vulnerabilities than those of Protocol III in the standard, because the protocols are not the same.

In this paper, we perform a security analysis on Ho's PAKE protocol, and show that it does not provide forward secrecy, although it is argued [8] that the protocol provides perfect forward secrecy. Furthermore, we show that the protocol is vulnerable to an impersonation attack, a KCI attack, and an invalid-curve attack. The impersonation attack on Ho's PAKE protocol is different from the impersonation attack on the corresponding PAKE protocol in the IEEE 802.15.6 standard [19]. By an invalid-curve attack, an adversary is able to extract the private key of another entity. The invalid-curve attack which is presented in this paper on Ho's PAKE protocol, is feasible by an insider adversary. However, it can be shown that any adversary can accomplish a similar invalid-curve attack on Ho's unauthenticated key exchange and numerical display AKE protocols. A variant of the impersonation attack, which is presented in this paper on Ho's PAKE protocol, is also feasible on Ho's AKE protocol with hidden public key transfer. Such extra vulnerabilities are due to not considering public key validations in Ho's protocols.

In this paper, we also perform a security analysis on Unger et al.'s PAKE protocol [25]. The protocol is a variant of the Ho's PAKE protocol, and is proposed for bridging the user interface gap in pervasive computing and smart environments. We show that Unger et al.'s PAKE protocol lacks forward secrecy, and is vulnerable to dictionary and replay attacks. The rest of this paper is organized as follows. We review the protocols in Section 2, and describe their vulnerabilities in Section 3.

2 Review of Two PAKE Protocols

Ho's PAKE protocol [8] and Unger et al.'s PAKE protocol [25] are depicted in Figures 1 and 2, respectively. They use public key cryptography on elliptic curves. The domain parameters consist of an elliptic curve E with cofactor h defined over the finite field $GF(p)$, where $p = q$ or 2^m in which q is a prime number of at least 160 bits, and m is larger than 160. The cofactor h of the elliptic curve is 1, 2 or 4. The base point G in the elliptic curve is of order n where $n \times G = O$ in which O denotes the point at infinity. There are other conditions that should be satisfied by domain parameters of elliptic curves in order to avoid known attacks on elliptic curve-based schemes [22], although they are not mentioned in [8, 25].

The protocols are executed between Alice ($\mathcal{A}$) and Bob ($\mathcal{B}$). $\mathcal{A}$ and $\mathcal{B}$ can be a node and a hub in a WBAN, respectively. $\mathcal{A}$ and $\mathcal{B}$ have self-generated public/private keys. It is specified neither in the IEEE 802.15.6 standard [1] nor in [8] if public keys are accompanied by digital certificates. However, it has been mentioned in [8] that "one of the two parties is likely to be severely constrained by memory, speed, or/and power, and hence cannot store public key certificates or perform digital signature calculations."

The private keys shall be 256-bit random integers, chosen independently from the set of integers $[1, n-1]$. The private key of $\mathcal{A}$ and $\mathcal{B}$ is denoted by SK_A and SK_B, respectively. The corresponding public keys are gen-

Alice ($\mathcal{A}$)		**Bob ($\mathcal{B}$)**
$SK_A \in_R [1, n-1]$		$SK_B \in_R [1, n-1]$
$PK_A = (PK_{AX}, PK_{AY}) = SK_A \times G$		$PK_B = (PK_{BX}, PK_{BY}) = SK_B \times G$
$PK'_A = PK_A - Q(PW)$		
Update SK_A and PK_A if $PK'_A = O$		
Select random 128-bit $Nonce_A$		Select random 128-bit $Nonce_B$
	$\xrightarrow{ID_B \| ID_A \| Nonce_A \| PK'_{AX} \| PK'_{AY} \| Other_A}$	
	$\xleftarrow{ID_A \| ID_B \| Nonce_B \| PK_{BX} \| PK_{BY} \| Other_B}$	
		$PK_A = PK'_A + Q(PW)$
$DH_{Key} = X(SK_A \times PK_B)$		$DH_{Key} = X(SK_B \times PK_A)$
$Temp_1 = RMB_{128}(DH_{Key})$		$Temp_1 = RMB_{128}(DH_{Key})$
$KMAC_{3A} = CMAC(Temp_1, ID_A \| ID_B \| Nonce_A \| Nonce_B \| Other_A, 64)$		$KMAC_{3B} = CMAC(Temp_1, ID_A \| ID_B \| Nonce_A \| Nonce_B \| Other_A, 64)$
$KMAC_{4A} = CMAC(Temp_1, ID_B \| ID_A \| Nonce_B \| Nonce_A \| Other_B, 64)$		$KMAC_{4B} = CMAC(Temp_1, ID_B \| ID_A \| Nonce_B \| Nonce_A \| Other_B, 64)$
	$\xleftarrow{ID_A \| ID_B \| Nonce_B \| PK_{BX} \| PK_{BY} \| Other_B \| KMAC_{3B}}$	
Halt if $KMAC_{3A} \neq KMAC_{3B}$		
	$\xrightarrow{ID_B \| ID_A \| Nonce_A \| PK'_{AX} \| PK'_{AY} \| Other_A \| KMAC_{4A}}$	
		Halt if $KMAC_{4A} \neq KMAC_{4B}$
$Temp_2 = LMB_{128}(DH_{Key})$		$Temp_2 = LMB_{128}(DH_{Key})$
$MK = CMAC(Temp_2, Nonce_A \| Nonce_B, 128)$		$MK = CMAC(Temp_2, Nonce_A \| Nonce_B, 128)$

Figure 1: Ho's PAKE protocol [8]

Alice ($\mathcal{A}$)		**Bob ($\mathcal{B}$)**
$SK_A \in_R [1, n-1]$		$SK_B \in_R [1, n-1]$
$PK_A = (PK_{AX}, PK_{AY}) = SK_A \times G$		$PK_B = (PK_{BX}, PK_{BY}) = SK_B \times G$
$PK'_A = PK_A - Q(PW)$		
Select random $Nonce_A$		
	$\xrightarrow{PK'_A, Nonce_A, ID_A, ID_B}$	
		Select random $Nonce_B$
		$PK_A = PK'_A + Q(PW)$
		$S = SK_B \times PK_A$
		$H_B = CMAC(PW, Other_B \| PK_B)$
	$\xleftarrow{PK_B, Nonce_B, ID_A, ID_B, H_B}$	
$S = SK_A \times PK_B$		
$H'_B = CMAC(PW, Other_B \| PK_B)$		
Verify if $H_B = H'_B$		
$H_A = CMAC(PW, Other_A \| PK_A)$		
$MK = CMAC(S, Nonce_A \| Nonce_B)$		
	$\xrightarrow{H_A}$	
		$H'_A = CMAC(PW, Other_A \| PK_A)$
		Verify if $H_A = H'_A$
		$MK = CMAC(S, Nonce_A \| Nonce_B)$

Figure 2: Unger et al.'s PAKE protocol [25]

erated as $PK_A = (PK_{AX}, PK_{AY}) = SK_A \times G$, and $PK_B = (PK_{BX}, PK_{BY}) = SK_B \times G$. PK_A and PK_B are points on E, and have X-coordinate and Y-coordinate values. The lifecycle of private/public keys are not specified in [8]. Although the key generation is depicted on the top of Figure 1, it does not mean that the key generation should repeat for each protocol execution. It is argued in [8] that all the proposed AKE and PAKE protocols provide the perfect forward secrecy. Reasoning for forward secrecy of the AKE protocols means that private/public keys are not random numbers used in a typical Diffie-Hellman key agreement. The forward secrecy makes sense if there is a static secret value. Furthermore, the private keys are specifically differentiated from nonces in the protocols.

$\mathcal{A}$ and $\mathcal{B}$ are assumed to have a shared password in advance. During protocol executions, $\mathcal{B}$ sends his public key PK_B in clear, but $\mathcal{A}$ sends a password-scrambled public key PK'_A that is masked by a hash of password as $PK'_A = PK_A - Q(PW)$ in which PW is a positive integer, converted through a character encoding from the pre-shared password between $\mathcal{A}$ and $\mathcal{B}$ such that $0 \leq PW < p$. The $Q(.)$ function is a mapping which converts the integer PW to the point $Q(PW) = (Q_X, Q_Y)$ on the elliptic curve in which $Q_X = 2^{32} \times 2h \times PW + M_X$ where M_X is the smallest nonnegative integer such that Q_X becomes the X-coordinate of a point on the elliptic curve. Q_Y is an even positive integer, and is the Y-coordinate of that point. $\mathcal{A}$ shall choose a private key SK_A such that the X-coordinate of PK_A is not equal to the X-coordinate of $Q(PW)$, i.e. we have $PK'_A \neq O$.

$CMAC(K, M, L)$ represents the L-bit output of the *Cipher-based Message Authentication Code* (CMAC), applied under key K to message M. $LMB_L(S)$ and $RMB_L(S)$ designates the L leftmost and the L rightmost bits of the bit string S, respectively. $X(P)$ denotes the X-coordinate of point P on the elliptic curve, i.e. $X(P) = X(P_X, P_Y) = P_X$. The sign $||$ denotes concatenation of bit strings. ID_A and ID_B may be MAC address, IP address, and so on. $Other_A$ and $Other_B$ denotes other public parameters of $\mathcal{A}$ and $\mathcal{B}$, respectively.

3 Security Analysis

In this section, we show that Ho's [8] and Unger et al.'s [25] PAKE protocols that are depicted in Figures 1 and 2, are vulnerable to different attacks. In the rest of this paper, $\mathcal{M}$ denotes an active adversary, and $\mathcal{E}$ denotes a passive adversary.

3.1 Security Problems of Ho's PAKE Protocol

It is argued that the Ho's PAKE protocol provides perfect forward secrecy, and is resilient to impersonation and dictionary attacks [8]. However, we show that the protocol lacks the forward secrecy, and is vulnerable to an impersonation attack, a KCI attack, and an invalid-curve attack.

3.1.1 Impersonation Attack

As mentioned in Section 2, public keys are self-generated by involved parties. It is more likely that public keys are not accompanied by digital certificates due to resource constraints on nodes. As neither $\mathcal{A}$ nor $\mathcal{B}$ checks the validity of the received public key,

- For impersonating $\mathcal{A}$, $\mathcal{M}$ can simply send O as the masked public key of $\mathcal{A}$. If $PK'_A = O$, then $DH_{Key} = O$.
- For impersonating $\mathcal{B}$, $\mathcal{M}$ can simply send O as the public key of $\mathcal{B}$. If $PK_B = O$, then $DH_{Key} = O$.

Based on the encoding used for representation of O, $Temp_1$ and $Temp_2$ will have a known value. The only secret information in calculation of $KMAC_{3A}$, $KMAC_{4A}$, $KMAC_{3B}$, and $KMAC_{4B}$ is $Temp_1$. As $Temp_1$ will have a known value, $\mathcal{M}$ can calculate $KMAC_{3A} = KMAC_{3B}$ and $KMAC_{4A} = KMAC_{4B}$, and bypass the authentication. The only secret information in calculation of the master key MK is $Temp_2$. As $Temp_2$ will have a known value, $\mathcal{M}$ can calculate MK. Then, $\mathcal{M}$ can successfully impersonate either $\mathcal{A}$ or $\mathcal{B}$.

Validation of a public key $PK = (PK_X, PK_Y)$ includes checking the following conditions [6, 15]:

1) $PK \neq O$,
2) $PK_X, PK_Y \in GF(p)$,
3) PK_X, PK_Y should satisfy the defining equation of curve E,
4) $h \times PK \neq O$ where h denotes the cofactor of E.

3.1.2 Key Compromise Impersonation Attack

The KCI attack is a weaker variant of the impersonation attack in terms of requiring knowledge of a private key for

kind of impersonation. The KCI attack is according to a stronger notion of security which has been considered in the eCK security model [10] for AKE protocols. If the private key of an entity $\mathcal{A}$ is compromised, an adversary $\mathcal{M}$ can impersonate $\mathcal{A}$ in one-factor authentication protocols. However, such compromise should not enable $\mathcal{M}$ to impersonate other honest entities in communication with $\mathcal{A}$. Resistance to the KCI attack is an important security attribute which prevents an adversary from actively controlling a compromised entity [23]. The KCI attack makes sense for PAKE protocols if they use public keys.

As Ho's PAKE protocol is vulnerable to an impersonation attack, one would consider discussion on the KCI attack redundant, because the KCI attack has an extra requirement for compromise of a private key. However, discussion on the KCI attack is noteworthy, because the impersonation attack on the protocol could be prevented by adding validation of public keys to the protocol. However, the KCI attack will be feasible even after adding public key validation or having certified public keys from a lightweight PKI [14, 21]. Here is the attack scenario in which $\mathcal{M}$ has SK_A, and impersonates $\mathcal{B}$. $\mathcal{M}$ does not need to have the password PW. As the public key of $\mathcal{B}$ is sent in clear, we can assume that $\mathcal{M}$ has obtained PK_B by eavesdropping a previous protocol run.

- $\mathcal{A}$ selects a 128-bit random number $Nonce_A$, and sends $\{ID_B \mid\mid ID_A \mid\mid Nonce_A \mid\mid PK'_{AX} \mid\mid PK'_{AY} \mid\mid Other_A\}$ to $\mathcal{B}$. $\mathcal{M}$ hijacks the session, and tries to impersonate $\mathcal{B}$.

- $\mathcal{M}$ selects a random number $Nonce_M$, and sends $\{ID_A||ID_B||Nonce_M||PK_{BX}||PK_{BY}||Other_B\}$ to $\mathcal{A}$.

- $\mathcal{M}$ has SK_A. $\mathcal{M}$ computes $DH_{Key} = X(SK_A \times PK_B)$, $Temp_1 = RMB_{128}(DH_{Key})$, $KMAC_{3B} = CMAC(Temp_1, ID_A \mid\mid ID_B \mid\mid Nonce_A \mid\mid Nonce_M \mid\mid Other_A, 64)$, and $KMAC_{4B} = CMAC(Temp_1, ID_B \mid\mid ID_A \mid\mid Nonce_M \mid\mid Nonce_A \mid\mid Other_B, 64)$. $\mathcal{M}$ sends $\{ID_A \mid\mid ID_B \mid\mid Nonce_M \mid\mid PK_{BX} \mid\mid PK_{BY} \mid\mid Other_B \mid\mid KMAC_{3B}\}$ to $\mathcal{A}$.

- $\mathcal{A}$ computes $DH_{Key} = X(SK_A \times PK_B)$, $Temp_1 = RMB_{128}(DH_{Key})$, and $KMAC_{3A} = CMAC(Temp_1, ID_A \mid\mid ID_B \mid\mid Nonce_A \mid\mid Nonce_M \mid\mid Other_A, 64)$. $\mathcal{A}$ verifies that $KMAC_{3A} = KMAC_{3B}$, and computes $KMAC_{4A} = CMAC(Temp_1, ID_B \mid\mid ID_A \mid\mid Nonce_M \mid\mid Nonce_A \mid\mid Other_B, 64)$. $\mathcal{A}$ sends $\{ID_B \mid\mid ID_A \mid\mid Nonce_A \mid\mid PK'_{AX} \mid\mid PK'_{AY} \mid\mid Other_A \mid\mid KMAC_{4A}\}$ to $\mathcal{M}$.

- $\mathcal{A}$ computes $Temp_2 = LMB_{128}(DH_{Key})$, and generates the master key $MK = CMAC(Temp_2, Nonce_A \mid\mid Nonce_M, 128)$.

- $\mathcal{M}$ computes $Temp_2 = LMB_{128}(DH_{Key})$, and generates the master key $MK = CMAC(Temp_2, Nonce_A \mid\mid Nonce_M, 128)$.

$\mathcal{M}$ and $\mathcal{A}$ compute the same MK. $\mathcal{M}$ could successfully impersonate $\mathcal{B}$.

3.1.3 Invalid-curve Attack

In Ho's protocols, neither $\mathcal{A}$ nor $\mathcal{B}$ consider validation of public keys, received from the other party. Validation of static and ephemeral public keys is very important in elliptic curve cryptography. An invalid-curve attack would be feasible if an EC-based protocol does not consider validation of static or ephemeral public keys [6, 20]. By an invalid-curve attack, an attacker may extract the private key of another entity [24].

In [8], the elliptic curve is defined over $GF(p)$ where $p = q$ or 2^m. For an elliptic curve defined over a finite field $GF(q)$ of prime order $q > 3$, the Weierstrass equation is $y^2 = x^3 + ax + b$ where $a, b \in GF(q)$. For non-singularity, we require that $4a^3 + 27b^2 \neq 0 \mod q$. For the binary finite fields $GF(2^m)$, the Weierstrass equation is $y^2 + xy = x^3 + ax^2 + b$ where $a, b \in GF(2^m)$ with $b \neq 0$. There is another kind of Weierstrass equation over $GF(2^m)$ which gives supersingular curves, but they are cryptographically weak. If G, the base point of the elliptic curve, is of order n, then h the cofactor of the elliptic curve is defined as $h = \#E(GF(p))/n$ in which $\#E(.)$ is called the order of the elliptic curve E, and it denotes the number of points on the elliptic curve (including O).

The idea behind an invalid-curve attack is that for two elliptic curves over $GF(q)$ (or two curves over $GF(2^m)$) whose defining equations have the same a coefficient but different b coefficients, the addition formulaes are the same, and they do not involve the coefficient b. For the general case, let $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ be the generalized Weierstrass equation of an elliptic curve E defined over the finite field $GF(p)$ where $p = q$ or 2^m. An *invalid-curve* (relative to E) is an elliptic curve E' defined over $GF(p)$ with the Weierstrass equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a'_6$ where $a_6 \neq a'_6$. Note that $E(GF(p)) \cap E'(GF(p)) = O$.

If $PK_M \in E'(GF(p))$ and $PK_M \neq O$, then there is not any private key SK_M such that $PK_M = SK_M \times G$. The addition formulaes on E and E' does not involve a_5

and a'_5 coefficient, respectively. Let $PK_M \in E'(GF(p))$, and SK_B be the private key of $\mathcal{B}$. If $\mathcal{B}$ uses the addition formulae for E in any point multiplication algorithm for computing $K = SK_B \times PK_M$, then $\mathcal{B}$ will indeed obtain a point on E', and we have $K \in E'(GF(p))$. If PK_M is a point of a small order, and $\mathcal{M}$ receives a feedback from $\mathcal{B}$ which includes some calculations based on K, this would be used for finding SK_B, the private key of $\mathcal{B}$.

For Ho's PAKE protocol, the invalid-curve attack can be accomplished by an insider adversary that has knowledge of the shared password PW. However, the same attack can be done by any adversary on Ho's unauthenticated key exchange protocol (Protocol I) and display AKE protocol (Protocol IV). Here is the attack scenario on Ho's PAKE protocol in which $\mathcal{A}$ performs an invalid-curve attack against $\mathcal{B}$, and finds SK_B:

- $\mathcal{A}$ selects an invalid curve E' such that $E'(GF(p))$ contains a point $PK_{Ai} = (PK_{AiX}, PK_{AiY})$ of small order t_i. $\mathcal{A}$ computes $PK'_{Ai} = (PK'_{AiX}, PK'_{AiY}) = PK_{Ai} - Q(PW)$, selects a random number $Nonce_A$, and sends $\{ID_B \,||\, ID_A \,||\, Nonce_A \,||\, PK'_{AiX} \,||\, PK'_{AiY} \,||\, Other_A\}$ to $\mathcal{B}$. Note that PK'_{Ai} most likely resides on neither E nor E'.

- $\mathcal{B}$ selects a 128-bit random number $Nonce_B$, and sends $\{ID_A \,||\, ID_B \,||\, Nonce_B \,||\, PK_{BX} \,||\, PK_{BY} \,||\, Other_B\}$ to $\mathcal{A}$.

- $\mathcal{B}$ computes $PK_{Ai} = PK'_{Ai} + Q(PW)$, $DH_{Key} = X(SK_B \times PK_{Ai})$, $Temp_1 = RMB_{128}(DH_{Key})$, $KMAC_{3B} = CMAC(Temp_1, ID_A \,||\, ID_B \,||\, Nonce_A \,||\, Nonce_B \,||\, Other_A,\ 64)$, and $KMAC_{4B} = CMAC(Temp_1,\ ID_B \,||\, ID_A \,||\, Nonce_B \,||\, Nonce_A \,||\, Other_B,\ 64)$. $\mathcal{B}$ sends $\{ID_A \,||\, ID_B \,||\, Nonce_B \,||\, PK_{BX} \,||\, PK_{BY} \,||\, Other_B \,||\, KMAC_{3B}\}$ to $\mathcal{A}$.

- $\mathcal{A}$ receives $KMAC_{3B}$, and halts the protocol execution. There are t_i possible values for $SK_B \times PK_{Ai}$ because PK_{Ai} is of order t_i. Then, there are $t_i/2$ possible values for $DH_{Key} = X(SK_B \times PK_{Ai})$ and $Temp_1 = RMB_{128}(DH_{Key})$. $\mathcal{A}$ tries all possible values of $Temp_1$ in $KMAC_{3A} = CMAC(Temp_1, ID_A \,||\, ID_B \,||\, Nonce_A \,||\, Nonce_B \,||\, Other_A,\ 64)$ until she finds a value for $Temp_1$ for which $KMAC_{3A} = KMAC_{3B}$. Then, with at most $t_i/2$ trials, $\mathcal{A}$ finds an equation $d_i^2 \equiv SK_B^2 \bmod t_i$ in which t_i and d_i are known, and SK_B is unknown.

$\mathcal{A}$ repeats the above attack for different points PK_{Ai} of pairwise relatively prime order t_i, i.e. we should have $gcd(t_i, t_j) = 1$, $\forall t_i \neq t_j$. Such points can be selected from the same or different invalid curves. Using the *Chinese remainder theorem*, $\mathcal{A}$ combines the mentioned equations, and finds $SK_B^2 \equiv d \mod N$ for some $N > n^2$. Since $SK_B^2 < n^2 < N$, we have $d = SK_B^2$, and $\mathcal{A}$ computes $SK_B = \sqrt{d}$. $\mathcal{A}$ finds the private key of $\mathcal{B}$, while $\mathcal{B}$ is unaware that such an attack has taken place.

3.1.4 Lack of Forward Secrecy

Forward secrecy is an important security attribute in key exchange protocols. If an entity's private key has been compromised, it should not affect the security of session keys that have been established before the compromise. The notion of *perfect forward secrecy* (PFS) is a bit stronger than the forward secrecy. PFS means that the established session keys should remain secure even after compromising the private keys of all the entities that are involved in the protocol. For public key-based AKE protocols, the forward secrecy is defined with respect to compromise of the private key. For PAKE protocols, the forward secrecy is defined with respect to compromise of the password. For PAKE protocols that use both public keys and passwords, the forward secrecy can be defined according to compromise of either a private key or a password.

In [8], it is argued for the PFS. However, we show that the protocol provides neither PFS nor forward secrecy. As PK_B, $Nonce_A$ and $Nonce_B$ are sent in clear, we can assume that they are eavesdropped and saved by $\mathcal{E}$. If SK_A is compromised, $\mathcal{E}$ computes $DH_{Key} = X(SK_A \times PK_B)$, $Temp_2 = LMB_{128}(DH_{Key})$, and obtains the master key $MK = CMAC(Temp_2, Nonce_A || Nonce_B, 128)$. Then, the protocol does not provide the forward secrecy.

3.2 Security Problems of Unger et al.'s PAKE Protocol

In this section, we show that Unger et al.'s PAKE protocol lacks the forward secrecy, and is vulnerable to dictionary and replay attacks.

3.2.1 Dictionary Attacks

It is crucial for PAKE protocols to be resilient to dictionary attacks. A PAKE protocol should not provide an adversary with a verifier which can be used for guessing the password. This is not the case for Unger et al.'s protocol. For an offline dictionary attack, it is sufficient for an adversary $\mathcal{E}$ to eavesdrop on messages exchanged between $\mathcal{A}$ and $\mathcal{B}$ in a protocol run. $\mathcal{E}$ then obtains H_B and PK_B that are sent

in clear. As values of $H_B = CMAC(PW, Other_B||PK_B)$, PK_B and $Other_B$ are known, it can be used as a verifier in a password guessing attack. $\mathcal{E}$ can try most probable passwords from a dictionary of passwords in the verifier.

Alternatively, instead of eavesdropping on a protocol execution, an adversary may interact with an entity, and obtain the verifier. Here is a scenario in which an adversary $\mathcal{M}$ impersonates $\mathcal{A}$, and obtains a verifier which can be used for finding the password:

- $\mathcal{M}$ selects random numbers SK_M and $Nonce_M$. $\mathcal{M}$ computes $PK''_M = SK_M \times G$, and sends $\{PK''_M, Nonce_M, ID_A, ID_B\}$ to $\mathcal{B}$.
- $\mathcal{B}$ selects a random number $Nonce_B$, computes $PK_M = PK''_M + Q(PW)$, $S = SK_B \times PK_M$, and $H_B = CMAC(PW, Other_B||PK_B)$. $\mathcal{B}$ sends $\{PK_B, Nonce_B, ID_A, ID_B, H_B\}$ to $\mathcal{M}$.
- $\mathcal{M}$ gets H_B, and halts the protocol execution. $\mathcal{M}$ performs an offline dictionary attack to find a password which satisfies $H_B = CMAC(PW, Other_B||PK_B)$ in which H_B, PK_B and $Other_B$ are known. $\mathcal{B}$ does not detect any attack or suspicious activity.

3.2.2 Replay Attack

Unger et al.'s protocol is vulnerable to a replay attack. In the protocol, authentication of $\mathcal{A}$ and $\mathcal{B}$ is done through $H_A = CMAC(PW, Other_A||PK_A)$ and $H_B = CMAC(PW, Other_B||PK_B)$, respectively. H_A and H_B does not contain any fresh information, and they will be the same in all protocol executions between $\mathcal{A}$ and $\mathcal{B}$, as long as they do not change their public keys or passwords. For the resource-constrained situation that has been considered in [25], $\mathcal{A}$ and $\mathcal{B}$ may perform private/public key generation once. Even if they change their public key, $\mathcal{B}$ always sends his public key PK_B in clear. $\mathcal{A}$ sends a password-scrambled public key PK'_A. If the values sent for PK'_A are different in different protocol runs, it notifies a change in PK_A or PW. Otherwise, it means that they are more likely unchanged which indicates feasibility of a replay attack. Of course, $\mathcal{M}$ cannot establish a new master key MK through a replay attack, but can bypass the authentication.

3.2.3 Lack of Forward Secrecy

As PK_B, $Nonce_A$ and $Nonce_B$ are sent in clear, we can assume that they are eavesdropped and saved by $\mathcal{E}$. If SK_A is compromised, $\mathcal{E}$ computes $S = X(SK_A \times PK_B)$, and obtains the master key $MK = CMAC(S, Nonce_A||Nonce_B)$. Then, the protocol does not provide the forward secrecy.

4 Conclusion

In this paper, we performed a security analysis on Ho's PAKE protocol [8] and Unger et al.'s PAKE protocol [25] that are proposed for body area networks and smart environments, respectively. Both protocols use elliptic curve cryptography. We showed that Ho's PAKE protocol lacks the forward secrecy and is vulnerable to impersonation, KCI and invalid-curve attacks. Furthermore, we showed that Unger et al.'s protocol lacks the forward secrecy, and is vulnerable to dictionary and replay attacks. The invalid-curve attack, which is presented in this paper on Ho's PAKE protocol, is feasible by an insider adversary where the adversary can extract the private key of another participant. However, it can be shown that any adversary can accomplish a similar invalid-curve attack on Ho's unauthenticated key exchange and numerical display AKE protocols [8]. A variant of the impersonation attack, which is presented in this paper on Ho's PAKE protocol, is also feasible on Ho's AKE protocol with hidden public key transfer [8]. Such extra vulnerabilities are due to not considering public key validations in Ho's protocols.

Acknowledgment

The author would like to thank anonymous reviewers for their comments.

References

[1] The IEEE Standards Association, "IEEE P802.15.6 Standard for Wireless Body Area Networks," 2012.

[2] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 72–84, 1992.

[3] Q. Cheng, "Cryptanalysis of a new efficient authenticated multiple-key exchange protocol from bilinear pairings," *International Journal of Network Security*, vol. 16, no. 6, pp. 494–497, 2014.

[4] Y. Ding and P. Horster, "Undetectable on-line password guessing attacks," *ACM SIGOPS Operating Systems Review*, vol. 29, pp. 77–86, Oct. 1995.

[5] R. Dutta and R. Barua, "Password-based encrypted group key agreement," *International Journal of Network Security*, vol. 3, no. 1, pp. 23–34, 2006.

[6] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to elliptic curve cryptography*, Springer, 2004.

[7] D. He, Y. Zhang, and J. Chen, "Cryptanalysis of a three-party password-based authenticated key exchange protocol," *International Journal of Network Security*, vol. 16, no. 5, pp. 393–396, 2014.

[8] J. M. Ho, "A versatile suite of strong authenticated key agreement protocols for body area networks," in *IEEE 8th International Conference on Wireless Communications and Mobile Computing (IWCMC'12)*, pp. 683–688, 2012.

[9] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," in *Advances in Cryptology (CRYPTO'05)*, pp. 546–566, 2005.

[10] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, pp. 1–16, 2007.

[11] C. C. Lee, S. T. Chiu, and C. T. Li, "Improving security of a communication-efficient three-party password authentication key exchange protocol," *International Journal of Network Security*, vol. 17, no. 1, pp. 1–6, 2015.

[12] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[13] M. Mana, M. Feham, and B. A. Bensaber, "Trust key management scheme for wireless body area networks," *International Journal of Network Security*, vol. 12, no. 2, pp. 75–83, 2011.

[14] S. Misra, S. Goswami, C. Taneja, and A. Mukherjee, "Design and implementation analysis of a public key infrastructure-enabled security framework for ZigBee sensor networks," *International Journal of Communication Systems*, Article first published online: 10 NOV 2014.

[15] M. Toorani, "SMEmail - a new protocol for the secure e-mail in mobile environments," in *Proceedings of the IEEE Australian Telecommunications Networks and Applications Conference (ATNAC'08)*, pp. 39–44, 2008.

[16] M. Toorani, "Cryptanalysis of a new protocol of wide use for email with perfect forward secrecy," *Security and Communication Networks*, vol. 8, no. 4, pp. 694–701, 2015.

[17] M. Toorani, "Cryptanalysis of a protocol from FC'10," in *Proceedings of Financial Cryptography and Data Security*, Jan. 2015.

[18] M. Toorani, "On continuous after-the-fact leakage-resilient key exchange," in *Proceedings of the Second ACM Workshop on Cryptography and Security in Computing Systems (CS2'15)*, pp. 31–34, 2015.

[19] M. Toorani, "On vulnerabilities of the security association in the IEEE 802.15.6 standard," in *Proceedings of Financial Cryptography and Data Security Workshops – 1st Workshop on Wearable Security and Privacy (Wearable'15)*, Jan. 2015.

[20] M. Toorani and A. Beheshti, "Cryptanalysis of an efficient signcryption scheme with forward secrecy based on elliptic curve," in *Proceedings of 2008 IEEE International Conference on Computer and Electrical Engineering (ICCEE'08)*, pp. 428–432, 2008.

[21] M. Toorani and A. Beheshti, "LPKI - a lightweight public key infrastructure for the mobile environments," in *Proceedings of the 11th IEEE International Conference on Communication Systems (ICCS'08)*, pp. 162–166, Nov. 2008.

[22] M. Toorani and A. Beheshti, "A directly public verifiable signcryption scheme based on elliptic curves," in *Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC'09)*, pp. 713–716, 2009.

[23] M. Toorani and A. Beheshti, "An elliptic curve-based signcryption scheme with forward secrecy," *Journal of Applied Sciences*, vol. 9, no. 6, pp. 1025–1035, 2009.

[24] M. Toorani and A. Beheshti, "Cryptanalysis of an elliptic curve-based signcryption scheme," *International Journal of Network Security*, vol. 10, no. 1, pp. 51–56, 2010.

[25] S. Unger and D. Timmermann, "Bridging the UI gap for authentication in smart environments," in *Proceedings of the 19th IEEE Symposium on Computers and Communications (ISCC'14)*, pp. 1–6, July 2014.

Mohsen Toorani received the B.S. degree in Communications Engineering and M.S. degree in Secure Communications both from Iran University of Science and Technology in 2005 and 2008, respectively. He is a PhD candidate at Department of Informatics, University of Bergen. His research interests include cryptology and information security.