THE PAIRING PROBLEM WITH USER INTERACTION

Thomas Peyrin
EPFL
Lausanne, Switzerland
thomas.peyrin@gmail.com

Serge Vaudenay
EPFL
Lausanne, Switzerland
serge.vaudenay@epfl.ch

Abstract Bluetooth-like applications face the pairing problem: two devices want to establish a relationship between them without any prior private information. Hoepman studied the ephemeral pairing problem by regarding the human operator of the devices as a messenger in an authenticated and/or private low-bandwidth channel between the nodes. Here we study the pairing problem with user interaction in which the operator can participate by doing extra (simple) computations.

Keywords: Authentication, pairing, key exchange

1. Introduction

A typical problem in wireless networks is that we do not know if two communicating devices are actually talking to each other. The pairing problem consists of securely establishing a private key between two or more specific physical nodes in the network. We assume that no secret information is shared between the nodes before the pairing. Furthermore, we want a high level of security and a minimal human interaction. Pairing between Bluetooth devices is a typical setting. In Hoe04, Hoepman studied the ephemeral pairing problem (denoted φKE): given a low bandwidth authentic and/or private communication channel between two nodes (called Alice and Bob), and a high bandwidth broadcast channel, can we establish a high-entropy shared secret session key without relying on any a priori shared secret information? The low bandwidth channel can be a (passive) human user who can read a PIN code on one de-

vice and write it on the other in a secure way. However there are many cases where this model is not sufficient: first, the standard Bluetooth pairing in which the user *generates* the PIN code; second, cases where the devices have no input keyboard or no output screen; third, when confidentiality (for instance) is guaranteed from the user to one device but not the other; etc. In this paper, we extend the model by introducing the user as a real participant who can further do simple computations. We call it the user-aided key exchange (UAKE) problem.

Gehrmann and Nyberg gave in GN01 two schemes. They also created a new scheme in GN04 using a MAC function and Jakobsson provided a variant of this scheme in Jak01. Those schemes are adapted to cases where one device has no input keyboard or no output screen.

The pairing problem is highly related to the authenticated key exchange problem (AKE): two users want to establish an authenticated high-entropy private key from scratch. Bellovin and Merritt BM92 gave a class of protocols called EKE (Encrypted Key Exchange) that solves the AKE problem using the assumption that the two peers already share a low-entropy password. EKE is basically an encrypted Diffie-Hellman DH76 key exchange. Jaspan Jas96 analyzed the Diffie-Hellman parameters in order to avoid partition attacks against EKE (in the case where the password is not ephemeral). Then Boyko *et al.* BMP00 specified a slightly different version of Diffie-Hellman based EKE called PAK (Password Authenticated Key exchange). MacKenzie Mac02 provided proofs in the Bellare-Rogaway model BR94. (A survey on authenticated key establishment protocols is available in BM03.) Note that in this paper, "EKE protocol" denotes independently the EKE or PAK protocol.

2. The pairing problem models

2.1 The pairing problems

In the pairing problem, two nodes in a (wireless ad-hoc) network, that do not yet share any secret, want to establish a secure association. They may be able to exchange small amounts of information reliably and/or in a private way by being attended by a human operator. The ephemeral key exchange (φKE) problem considers the human operator as a simple messenger between the nodes. In this paper we consider the user-aided key exchange (UAKE) problem in which the operator really participates. The nodes can communicate through the insecure channel and the user can securely exchange small amounts of information with the nodes and perform simple operations. Protocols must be such that:

1 both nodes and the user are ensured that the secret is shared with the correct physical node

2 no other node learns any part of the shared secret

3 a user needs to perform only simple and intuitive steps

For the third requirement, we allow the following operations: pick a random string, compare two strings, copy a string, XOR two strings. Avoiding the (quite complicated) XOR will be addressed in Section 4.1. We further limit user channels to a small bandwidth. The second requirement will be made clear by formalizing the security model. Once achieved, the first requirement is satisfied by standard key confirmation techniques. Note that we do not consider denial of services attacks.

By directly introducing the user in the problem, we can consider many different situations that can be encountered in practice. For example, we can easily describe the Bluetooth pairing in many different scenarios such as devices with no output screen or no input keyboard, pairing in a hostile environment when anyone can look over the user's shoulder, etc.

2.2 The communication model

Two nodes Alice and Bob are connected through a high bandwidth channel network. The adversary Eve has full control over this channel. Both nodes however share with the user two communication channels (one in each direction) which can have specific security properties:

1 **confidentiality:** the sender is guaranteed that the messages she sends can not be read by anyone but the right receiver (Eve can not read it).

2 **integrity:** the receiver is guaranteed that the message he receives was actually sent as is (Eve can not modify it).

3 **authentication:** the receiver is guaranteed that the message he receives was actually sent by the right sender (Eve can not modify or insert a message in the channel but can delay or replay a message). (Note that our definition of authentication implicitly assumes integrity.)

These properties may hold in both directions, or only in one direction. In this paper, we will not consider the integrity property except in our final discussion in Section 4.1 to simplify the protocols. Note that lack of integrity protection in confidential channels means that it could be possible for Eve to replace a confidential z message by a message $z \oplus \delta$ with a δ of her choice. (This is typically the case when the confidential channel is implemented by a stream cipher, e.g. in Bluetooth.) We further assume independence between the channels in the sense that it is impossible for an adversary e.g. to take a message from a secure channel and to insert it into another.

We thus have 4 unidirectional channels that can have one of four attributes: *AC* (authenticated and confidential channel), *A* (authenticated channel), *C* (confidential channel) and 0 (no security property). Those channels represent all

the interactions with the user. For example, a screen on a device represents a channel of type A from the device to the user who is watching the screen, a device holder typing a code on the device's keyboard in a private way represents a channel of type AC from the user to the device. Moreover, we can consider channels with an extremely low bandwidth (typically one bit) if we use a single light, or a single Boolean button for low cost devices.

2.3 The security model

We use the adversary model of Bellare *et al.* BPR00. Each participant p may engage in the protocol many times in a concurrent way. For each new protocol run where p is asked to play a role, a unique instance π_p^i is created. Eve has the entire control of the network and about who is running a new step of a protocol run. In a UAKE protocol with participants p, q, and r playing the role of Alice, Bob, and User respectively, we create new instances π_p^i, π_q^j, and π_r^k with input (p,q,r). π_p^i and π_q^j should terminate with a key. (The φKE protocol is similar: r is simply hidden.) The attack is formalized by giving access to oracles for the instances of the network to the adversary:

- Execute$(\pi_p^i, \pi_q^j, \pi_r^k)$**:** execute a complete protocol run with π_p^i, π_q^j, and π_r^k. This query models passive attacks.

- Corrupt(p,x)**:** get all internal information about p and force its secret data (if any) to become x.

- Reveal(π_p^i)**:** reveal the key generated by π_p^i to the adversary.

- Send(π_p^i, m)**:** send a message m to the instance π_p^i and run a new step of the protocol.

- Test(π_p^i)**:** this query can be called only once. A bit b is flipped at random a random key (if $b = 0$) or the key from π_p^i (if $b = 1$) is output.

Eve makes a Test query and tries to correctly guess the bit b. The attack is successful if p,q,r are not corrupted and if Test(π_p^i) or Test(π_q^j) led to the right guess for b. Thus we define the advantage of Eve attacking the protocol by $Adv_E = 2\Pr[\text{correct}] - 1$. Note that we can not send a Test(π_p^i) query if a Reveal(π_p^i) or Reveal(π_q^j) query has already been sent, otherwise finding the value of the bit b would be trivial. We do not consider long term passwords as in regular EKE schemes but rather ephemeral ones. So oracles Reveal(π_p^i) and Corrupt(p,x) are not relevant in our context.

3. Key exchange with user interaction

3.1 The ephemeral pairing problem

In the original φKE problem, we have $2^4 = 16$ different possible configurations (2 channels and 4 possible security properties for each channel). We can represent each of those configurations by a 2×2 Boolean matrix: each row corresponds to a security property (A and C), and each column corresponds to a channel. For more readability, we represent the matrix by $M = [A \overset{a}{\underset{b}{\rightleftharpoons}} B]$ where $a, b \in \{0, A, C, AC\}$ are the columns of M. We denote $\varphi\text{KE}(M)$ the φKE problem with the configuration represented by the matrix M. If a secure protocol can be found for the $\varphi\text{KE}(M)$ problem, we say that $\varphi\text{KE}(M)$ is possible. Otherwise, we say that it is impossible. First of all, we can see that the φKE problem is symmetric: $\varphi\text{KE}(M)$ is equivalent to $\varphi\text{KE}(\text{sym}(M))$ where $\text{sym}(M)$ is the M matrix with the columns inverted. Furthermore, if a $\varphi\text{KE}(M_1)$ problem represented by the configuration matrix M_1 is possible, we can solve the problem with additional security properties by using the same protocol. We denote $M_1 \leq M_2$ for corresponding configuration matrices M_2.

FACT 1 *Let M_1 and M_2 be two φKE problem configuration matrices. If $M_1 \leq M_2$, any protocol which solves $\varphi KE(M_1)$ solves $\varphi KE(M_2)$ as well.*

FACT 2 *Let M be a φKE problem configuration matrix. $\varphi KE(M)$ is possible if and only if $\varphi KE(\text{sym}(M))$ is possible.*

THEOREM 3 (HOE05) *$\varphi KE(A \overset{C}{\underset{A}{\rightleftharpoons}} B)$ is impossible.*

Hoepman provided protocols for all minimal possible configurations. We can see that two types of protocols are used: we can try to make Alice and Bob share a low-entropy password and compute the EKE protocol with that password (see Figure 1 and Figure 2). The two devices can also try to run a Diffie-Hellman key exchange with commitment and authenticate with the low bandwidth channel (see Figure 3).

THEOREM 4 (HOE04) *$\varphi KE(A \xrightarrow{AC} B)$ and $\varphi KE(A \overset{C}{\underset{C}{\rightleftharpoons}} B)$ are possible by using the protocol from Figures 1 and 2 respectively. The advantage of an adversary which is limited to q oracles queries is at most the best advantage of an adversary to the EKE protocol with the same parameter q.*

THEOREM 5 (HOE04) *We consider a group G of order at least 2^{2s} in which the decisional Diffie-Hellman problem is hard. We consider five hash functions $h_1 : G \to G$, $h_2 : G \to \{0,1\}^t$, $h_3, h_4, h_5 : G \to \{0,1\}^\sigma$ such that $h_1(X)$ and $h_2(X)$ are independent for $X \in_U G$, h_2 is balanced, and h_3, h_4, and h_5 are*

(independent) pairwise independent random hash functions. $\varphi KE(A \underset{A}{\overset{A}{\rightleftharpoons}} B)$ *is possible by using the protocol from Figure 3. The advantage of an adversary which is limited to q oracle queries is* $O(1-e^{-q.2^{-t}})+O(2^{-s})$.

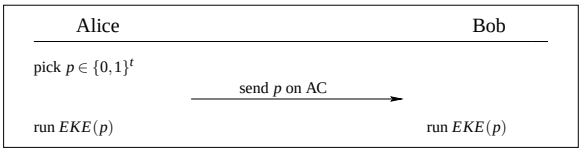

Figure 1. $\varphi\text{KE}(A \xrightarrow{AC} B)$

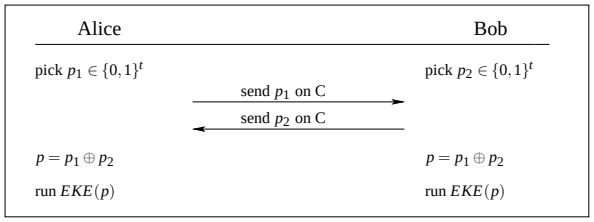

Figure 2. $\varphi\text{KE}(A \underset{C}{\overset{C}{\rightleftharpoons}} B)$

3.2 The user-aided key exchange problem

In the UAKE problem we have $4^4 = 256$ different possible configurations (4 channels and 4 different states for each channel). We can represent each of those configurations by a 2×4 matrix as in the φKE problem. For more readability, we represent the matrix by $M = [A \underset{b}{\overset{a}{\rightleftharpoons}} U \underset{c}{\overset{d}{\rightleftharpoons}} B]$ where $a,b,c,d \in \{0,A,C,AC\}$ correspond to the columns. We denote $\text{UAKE}(M)$ the UAKE problem with the configuration represented by the matrix M. The UAKE problem is symmetric: $\text{UAKE}(M)$ is the same problem as $\text{UAKE}(\text{sym}(M))$ where $\text{sym}(M)$ is the M matrix with some columns inverted so that the role of Alice and Bob is exchanged.

FACT 6 *Let* M_1 *and* M_2 *be two UAKE problem configuration matrices. If* $M_1 \leq M_2$*, any protocol solving* $UAKE(M_1)$ *solves* $UAKE(M_2)$ *as well.*

FACT 7 *Let M be a UAKE problem configuration matrix.* $UAKE(M)$ *is possible if and only if* $UAKE(\text{sym}(M))$ *is possible.*

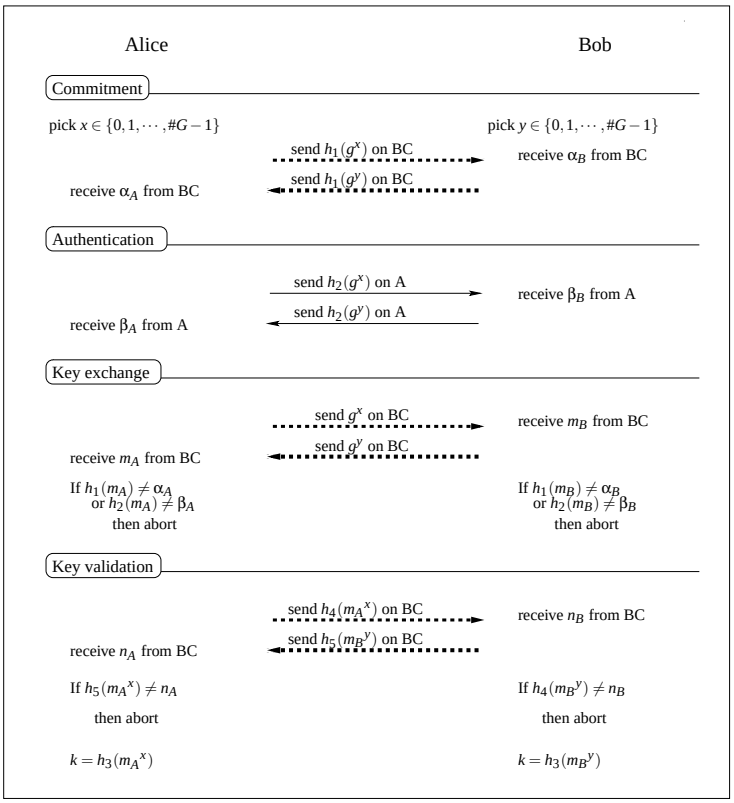

Figure 3. $\varphi\mathrm{KE}(A \underset{A}{\overset{A}{\rightleftharpoons}} B)$

We consider two participants Alice and Bob of $\mathrm{UAKE}(A \underset{b}{\overset{a}{\rightleftharpoons}} U \underset{c}{\overset{d}{\rightleftharpoons}} B)$ protocol. By simulating the interaction between Alice and User by a participant C. We obtain a protocol for $\varphi\mathrm{KE}(C \underset{c}{\overset{d}{\rightleftharpoons}} B)$. We deduce:

FACT 8 *Let $a,b,c,d \in \{0,A,C,AC\}$. If $UAKE(A \underset{b}{\overset{a}{\rightleftharpoons}} U \underset{c}{\overset{d}{\rightleftharpoons}} B)$ is possible then $\varphi KE(A \underset{b}{\overset{a}{\rightleftharpoons}} B)$ and $\varphi KE(A \underset{c}{\overset{d}{\rightleftharpoons}} B)$ are also possible.*

Considering a messenger U who forwards messages, we obtain:

FACT 9 *Let $a,b \in \{0,A,C,AC\}$. If $\varphi KE(A \underset{b}{\overset{a}{\rightleftharpoons}} B)$ is possible then $UAKE(A \underset{b}{\overset{a}{\rightleftharpoons}} U \underset{b}{\overset{a}{\rightleftharpoons}} B)$ is also possible.*

THEOREM 10 *Let* $a,b,c,d \in \{0,A,C,AC\}$. *UAKE*$(A \overset{a}{\underset{b}{\rightleftharpoons}} U \overset{d}{\underset{c}{\rightleftharpoons}} B)$ *is possible if and only if* φ*KE*$(A \overset{a}{\underset{b}{\rightleftharpoons}} B)$ *and* φ*KE*$(A \overset{d}{\underset{c}{\rightleftharpoons}} B)$ *are possible. Channels with security property* 0 *can be removed, except for* $(A \xrightarrow{AC} U \xleftarrow{AC} B)$ *which is impossible.*

Proof: Let us prove that UAKE$(A \xrightarrow{AC} U \xleftarrow{AC} B)$ is impossible. In that configuration, Alice and Bob can not receive anything from any secure channel. By removing any interaction with U, we obtain a φKE$(A \overset{0}{\underset{0}{\rightleftharpoons}} B)$ protocol which contradicts Theorem 3 and Fact 1. Other impossible cases follow from Fact 8.

Let us now show that UAKE$(A \overset{a}{\underset{b}{\rightleftharpoons}} U \overset{d}{\underset{c}{\rightleftharpoons}} B)$ is possible for all combinations of φKE limit cases: φKE$(A \overset{A}{\underset{A}{\rightleftharpoons}} B)$, φKE$(A \overset{C}{\underset{C}{\rightleftharpoons}} B)$, φKE$(A \xrightarrow{AC} B)$ and φKE$(A \xleftarrow{AC} B)$. By using symmetries, we restrict to the following limit cases:

- Type 1: $(A \xrightarrow{AC} U \xrightarrow{AC} B)$, $(A \overset{C}{\underset{C}{\rightleftharpoons}} U \overset{C}{\underset{C}{\rightleftharpoons}} B)$, $(A \overset{A}{\underset{A}{\rightleftharpoons}} U \overset{A}{\underset{A}{\rightleftharpoons}} B)$.
- Type 2: $(A \xleftarrow{AC} U \xrightarrow{AC} B)$, $(A \overset{C}{\underset{C}{\rightleftharpoons}} U \xrightarrow{AC} B)$, $(A \overset{C}{\underset{C}{\rightleftharpoons}} U \xleftarrow{AC} B)$.
- Type 3: $(A \overset{A}{\underset{A}{\rightleftharpoons}} U \overset{C}{\underset{C}{\rightleftharpoons}} B)$, $(A \overset{A}{\underset{A}{\rightleftharpoons}} U \xrightarrow{AC} B)$, $(A \overset{A}{\underset{A}{\rightleftharpoons}} U \xleftarrow{AC} B)$, $(A \overset{AC}{\underset{0}{\rightleftharpoons}} U \xleftarrow{AC} B)$.

Fact 9 addresses limit cases of type 1. Theorem 11 and 12 below provide a solution for limit cases of type 2 and 3 respectively. □

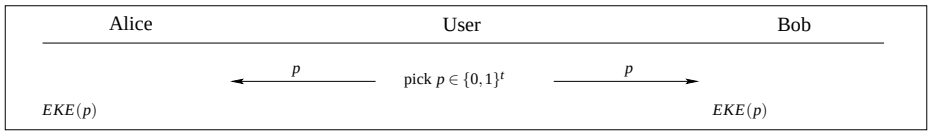

Figure 4. UAKE$(A \xleftarrow{AC} U \xrightarrow{AC} B)$

THEOREM 11 *UAKE*$(A \xleftarrow{AC} U \xrightarrow{AC} B)$, *UAKE*$(A \overset{C}{\underset{C}{\rightleftharpoons}} U \xrightarrow{AC} B)$ *and UAKE*$(A \overset{C}{\underset{C}{\rightleftharpoons}} U \xleftarrow{AC} B)$ *are possible by using the protocols from Figures 4, 5 and 6 respectively. The advantage of an adversary which is limited to q oracles queries is at most the best advantage of an adversary to the EKE protocol with the same parameter q plus* 2^{-t}.

Proof: The Figure 4 case is trivial: we assume we can set up a password in a secure way prior to EKE. For the cases of Figures 5 and 6, we note that if the

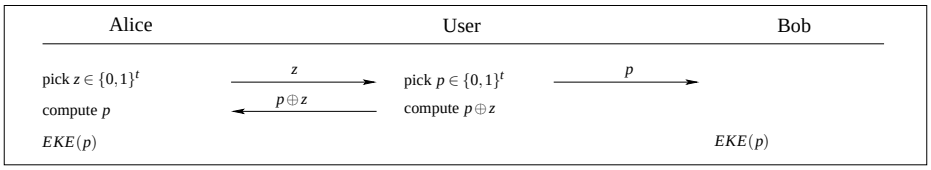

Figure 5. UAKE($A \underset{C}{\overset{C}{\rightleftharpoons}} U \xrightarrow{AC} B$)

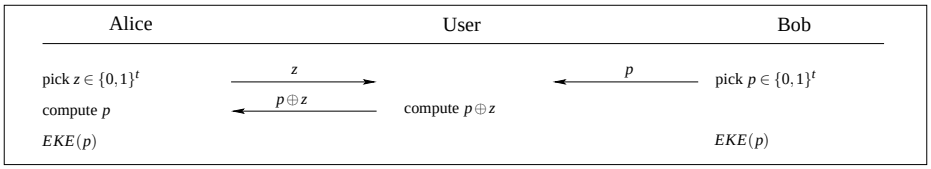

Figure 6. UAKE($A \underset{C}{\overset{C}{\rightleftharpoons}} U \xleftarrow{AC} B$)

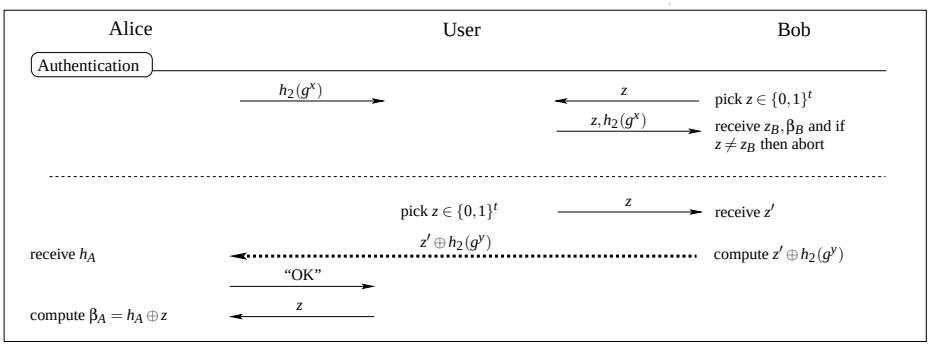

Figure 7. Authentication step in Figure 3 for UAKE($A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{C}{\overset{IC}{\rightleftharpoons}} B$)

adversary impersonates User to Alice, since she has no clue about (Bob's) p, Alice will receive an incorrect p with probability $1-2^{-t}$ and EKE will fail. □

THEOREM 12 *With the same hypotheses as in Theorem 5, UAKE*($A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{C}{\overset{IC}{\rightleftharpoons}} B$), *UAKE*($A \underset{A}{\overset{A}{\rightleftharpoons}} U \xrightarrow{AC} B$), *UAKE*($A \underset{A}{\overset{A}{\rightleftharpoons}} U \xleftarrow{AC} B$), *and UAKE*($A \underset{0}{\overset{AC}{\rightleftharpoons}} U \xleftarrow{AC} B$) *are possible by using the sub-protocols from Figures 7, 8, 9, and 10 respectively in the protocol of Figure 3. The advantage of an adversary which is limited to q oracle queries is $O(q.2^{-t}) + O(2^{-s})$. The first part of the protocol on Figure 7 further assumes integrity in the $U \rightarrow B$ channel.*

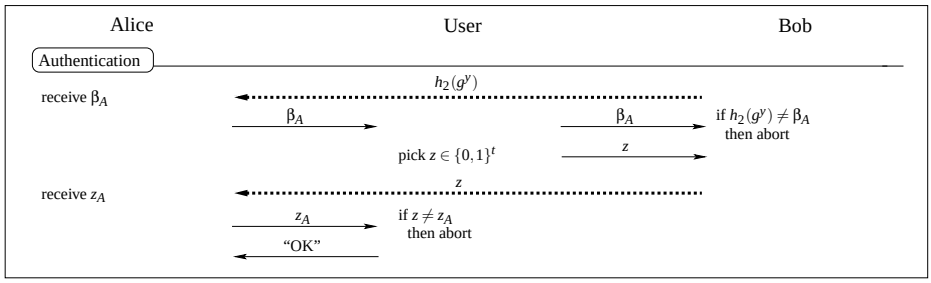

Figure 8. Authenticated channel from Bob to Alice in $A \overset{A}{\underset{A}{\rightleftharpoons}} U \xrightarrow{AC} B$

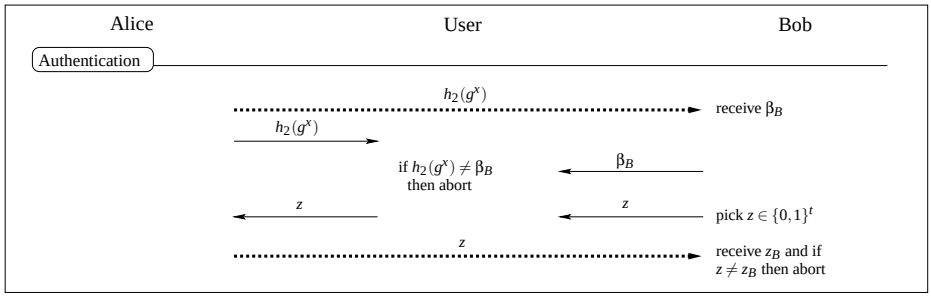

Figure 9. Authenticated channel from Alice to Bob in $A \overset{A}{\underset{A}{\rightleftharpoons}} U \xleftarrow{AC} B$

A heavier protocol for UAKE$(A \overset{A}{\underset{A}{\rightleftharpoons}} U \overset{C}{\underset{C}{\rightleftharpoons}} B)$ without the integrity assumption is provided in Section 4.5.

Proof (sketch): In Figure 8 (resp. Figure 9), if the adversary impersonates Bob to Alice (resp. Alice to Bob), the random z will never be released, so the protocol cannot succeed but with a probability of 2^{-t}. Figure 10 is similar.

In Figure 7 second part, the adversary has no clue about $h_2(g^y)$ and z until User discloses z. So, if she impersonates Bob to Alice, she can not predict which $h_2(g^y)$ Alice will obtain. Consistency check with the commitment phase in Hoepman's protocol will thus reject with a probability of $1-2^{-t}$ (Note that this works because $h_2(g^y)$ is unknown prior to the protocol).□

4. Discussions

4.1 Removing the XORs

We can see that the user has to compute the XOR of two values in protocols from Figures 5 and 6. Those cases have a common particularity: we have a

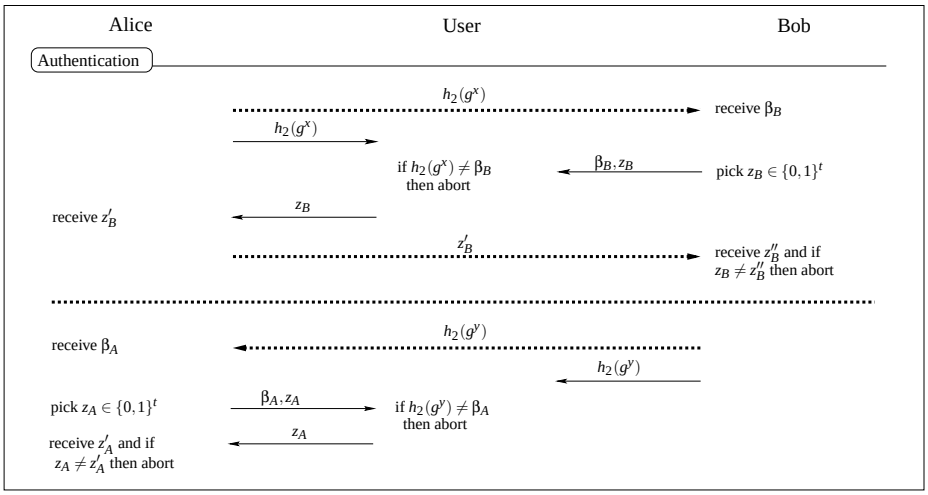

Figure 10. Authentication step in Figure 3 for UAKE($A \underset{0}{\overset{AC}{\rightleftharpoons}} U \overset{AC}{\longleftarrow} B$)

confidential non-authenticated channel between the user and Alice or Bob. In pairing situations those cases may not be relevant: such a channel would e.g. mean for example that the user types some digits on one device in a private way but the device is not sure that the typed digits actually come from the user! Nevertheless, those XORs can be removed by assuming integrity in addition to confidentiality. (Virtual confidential channels achieving this can typically be implemented by using encryption with strong security properties, e.g. IND-CCA2. Using less secure encryption, e.g. CBC encryption requires extra care.) In that case we replace the XORs by the concatenation.

4.2 User operations and bandwidth

One of the crucial points of our protocols are the ease of use for the user, we can thus analyze the number of operations computed by the user on the devices. In Figure 11 is shown the user operations according to the different protocols. According to the previous section, we can remove the XORs; in our table that would mean adding two copied values for each XOR.

4.3 Applications

A typical application of the UAKE problem would be the Bluetooth authentication scheme. The standard Bluetooth pairing assumes the same configuration as the protocol shown on Figure 4: the user types a password on both devices in a private and authenticate way. But according to our analysis of the UAKE

UAKE problems	pick	compare	copy	XOR	receive	send
$A \xrightarrow{AC} U \xrightarrow{AC} B$			t		t	t
$A \underset{C}{\overset{C}{\rightleftharpoons}} U \underset{C}{\overset{C}{\rightleftharpoons}} B$			2t		2t	2t
$A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{A}{\overset{A}{\rightleftharpoons}} B$			2t		2t	2t
$A \xleftarrow{AC} U \xrightarrow{AC} B$	t		t			2t
$A \underset{C}{\overset{C}{\rightleftharpoons}} U \xrightarrow{AC} B$	t			t	t	2t
$A \underset{C}{\overset{C}{\rightleftharpoons}} U \xleftarrow{AC} B$				t	2t	t
$A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{C}{\overset{IC}{\rightleftharpoons}} B$	t		3t		2t + 1	4t
$A \underset{A}{\overset{A}{\rightleftharpoons}} U \xrightarrow{AC} B$	t	t	t		3t	3t + 1
$A \underset{A}{\overset{A}{\rightleftharpoons}} U \xleftarrow{AC} B$		t	t		4t	2t
$A \underset{0}{\overset{AC}{\rightleftharpoons}} U \xleftarrow{AC} B$		2t	2t		6t	2t

Figure 11. User operations in UAKE protocols (in bits)

problem, we can consider many other cases. For example, the user can read a password on device Alice and copy it on the device Bob. Moreover, we can imagine as explained on Figure 9, that the device Bob has only a private and authenticate screen but no keyboard and that the user can read and type data on device Alice in an authenticated way but not in a private way.

Another application could be the establishment of a secure SSL or SSH session without certificates. In the $A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{A}{\overset{A}{\rightleftharpoons}} B$ case, the user could indeed be two human operators talking (in an authenticated way) over the telephone, i.e. a $A \underset{A}{\overset{A}{\rightleftharpoons}} U_A \underset{A}{\overset{A}{\rightleftharpoons}} U_B \underset{A}{\overset{A}{\rightleftharpoons}} B$ scenario.

Note that problems arise if we do not consider mutual belief in the key as shown by Lowe in Low96. The UAKE protocols should similarly be followed by an acknowledgment protocol.

4.4 Manufacturer aided key exchange

We can consider that a password p_M has been written in the non-volatile memory of one device by the manufacturer, for example for a low-cost device without any keyboard. That would mean a fourth node M in our pairing scheme representing the manufacturer. AC channels from M to Bob and to User can be considered. Note that those channels can only be used once in the first setup. That new assumption would change protocols shown on Figures 4, 5 and 6: we

use now p_M for the *EKE* protocol. This works in a $A \overset{C}{\leftarrow} U \overset{AC}{\longleftarrow} M \overset{AC}{\longrightarrow} B$ setting. Note that obviously this scheme leads to weaker versions of our protocols since the password used for each instance remains always the same.

We can also easily adapt the $\varphi\mathrm{KE}(A \underset{A}{\overset{A}{\rightleftharpoons}} B)$ protocol in Figure 3 to solve the pairing problem in a $A \underset{A}{\overset{A}{\rightleftharpoons}} U \overset{A}{\rightarrow} B$ or $A \underset{A}{\overset{A}{\rightleftharpoons}} U \overset{A}{\leftarrow} B$ configuration with a prior $U \overset{AC}{\longleftarrow} M \overset{AC}{\longrightarrow} B$ setup. We can even restrict one of the two $A \underset{A}{\overset{A}{\rightleftharpoons}} U$ channels to a single bit.

4.5 $\mathbf{UAKE}(A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{C}{\overset{C}{\rightleftharpoons}} B)$

We now consider the protocol on Figure 12 as a replacement for the authentication phase in the protocol of Figure 3 using $\mathrm{GF}(2^t)$ arithmetics.

In the first part of the protocol, we consider an optimal adversary who tries to make Bob accept a β of his choice for $X = h_2(g^x)$. (This follows a commitment phase in Hoepman's protocol, so an attack which makes Bob accept a random value is thwarted by the consistency check when opening the commitment.) Note that the right value of X is unknown to the adversary prior to the protocol. Without loss of generality, the adversary replaces (u,v) by $(u',v') = f(u,v)$, the returned (u',v') by $(u'',v'') = g(u',v')$, X by β, and $w = u' + v'\beta$ by $w' = h_X(w)$ for some chosen functions f, g, and h_X.

Let S_w be the set of all (u,v) such that $g(u',v') = (u,v)$ for $(u',v') = f(u,v)$, and $u' + v'\beta = w$. Note that $\#S_w \leq 2^t$. The attack is successful if and only if (X,u,v) is such that there exists w such that $u + vX = h_X(w)$ and $(u,v) \in S_w$. Hence the probability of success p is

$$p = \frac{1}{2^{2t}(2^t - 1)} \sum_w \sum_X \#\{(u,v) \in S_w; u + vX = h_X(w)\}.$$

Given w, let now n_i be the number of X's such that $\{(u,v) \in S_w; u + vX = h_X(w)\}$ has cardinality i. We can view the (u,v) pairs as straight lines. Given a set of i straight lines such that $u + vX = h_X(w)$ for one fixed X and w, we have $i(i-1)/2$ pairs of straight lines intersecting on the same point. If we sum all pairs over all X's, we obtain an overall number of intersecting pairs of at most $\#S_w \times (\#S_w - 1)/2$. Hence

$$\sum_i n_i \times \frac{i(i-1)}{2} \leq \frac{\#S_w \times (\#S_w - 1)}{2} \leq \frac{2^t(2^t - 1)}{2}.$$

We have

$$\sum_X \#\{(u,v) \in S_w; u + vX = h_X(w)\} = \sum_{i=1}^{2^t} i.n_i$$

with the constraint $\sum_i n_i \leq 2^t$. By linear programming results we obtain that

$$\sum_X \#\{(u,v) \in S_w; u + vX = h_X(w)\} \leq O\left(2^{3t/2}\right)$$

hence $p \leq O\left(2^{-t/2}\right)$. This big O is thus a new term to add in Theorem 12 for our protocol without the integrity assumption.

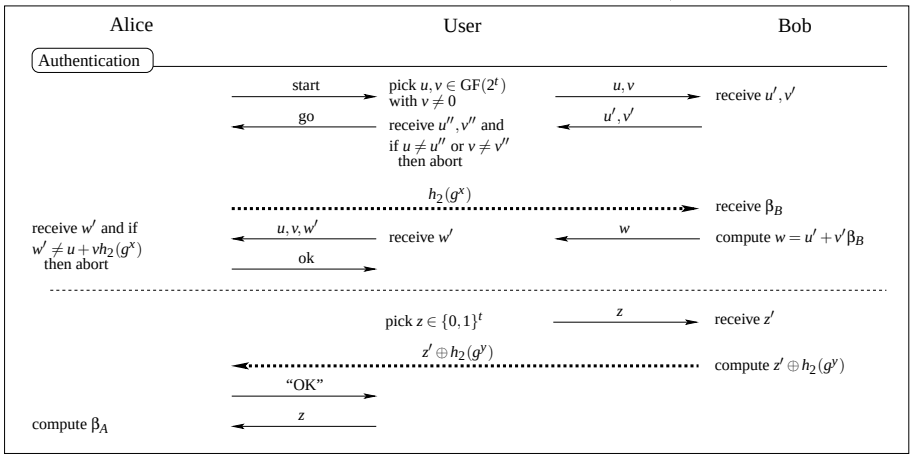

Figure 12. Authenticated step in Figure 3 for UAKE($A \underset{A}{\overset{A}{\rightleftharpoons}} U \underset{C}{\overset{C}{\rightleftharpoons}} B$)

5. Conclusion

We have extended Hoepman's ephemeral pairing problem by introducing the User Aided Key Exchange problem. We studied the minimal assumptions and provided pairing protocols in all cases.

Acknowledgment. This work was supported in part by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

References

Steven M. Bellovin and Michael Merritt, *Encrypted key exchange : Password-based protocols secure against dictionnary attacks*, IEEE symposium on Research in Security and Privacy, IEEE Computer Society Press, 1992, pp. 72–84.

Colin Boyd and Anish Mathuria, *Protocols for authentication and key establishment*, Information Security and Cryptography, Springer-Verlag, 2003.

Victor Boyko, Phillip MacKenzie, and Sarvar Patel, *Provably secure password-authenticated key exchange using Diffie-Hellman*, Advances in Cryptology (Eurocrypt'00), Lecture notes in computer science, vol. 1807, Springer-Verlag, 2000, pp. 156–171.

Mihir Bellare, David Pointcheval, and Phillip Rogaway, *Authenticated key exchange secure against dictionary attacks*, Advances in Cryptology (Eurocrypt'00), Lecture notes in computer science, vol. 1807, Springer-Verlag, 2000, pp. 139–155.

Mihir Bellare and Phillip Rogaway, *Entity authentication and key distribution*, Advances in Cryptology (Crypto'93), Lecture notes in computer science, vol. 773, Springer-Verlag, 1994, p. 232.

Whitfield Diffie and Martin E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT-22(6)** (1976), 644–654.

Christian Gehrmann and Kaisa Nyberg, *Enhancements to Bluetooth baseband security*, Proceedings of Nordsec 2001, 2001, Copenhagen, Denmark.

_____, *Security in personal area networks*, Security for Mobility (2004), 191–230, IEE, London.

Jaap-Henk Hoepman, *The ephemeral pairing problem*, Eighth International Conference on Financial Cryptography, Lecture notes in computer science, vol. 3110, Springer-Verlag, 2004, Key West, FL, USA, pp. 212–226.

_____, *Ephemeral pairing on anonymous networks*, To appear in the Proceedings of Second IEEE International Workshop on Pervasive Computing and Communication Security, Lecture notes in computer science, Springer-Verlag, 2005.

Markus Jakobsson, *Method and apparatus for immunizing against offline dictionary attacks*, U.S. Patent Application 60/283,996. Filed on 16th April 2001, 2001.

Barry Jaspan, *Dual-workfactor encrypted key exchange : Efficiently preventing password chaining and dictionnary attacks*, 6th USENIX Security Symposium, San Jose, California, 1996, pp. 43–50.

Gavin Lowe, *Some new attacks upon security protocols*, 9th IEEE Computer Security Fondations Workshop, IEEE Computer Society Press, 1996, pp. 162–169.

Philip MacKenzie, *The PAK suite : Protocols for password-authenticated key exchange*, Tech. Report 2002-46, DIMACS, 2002.