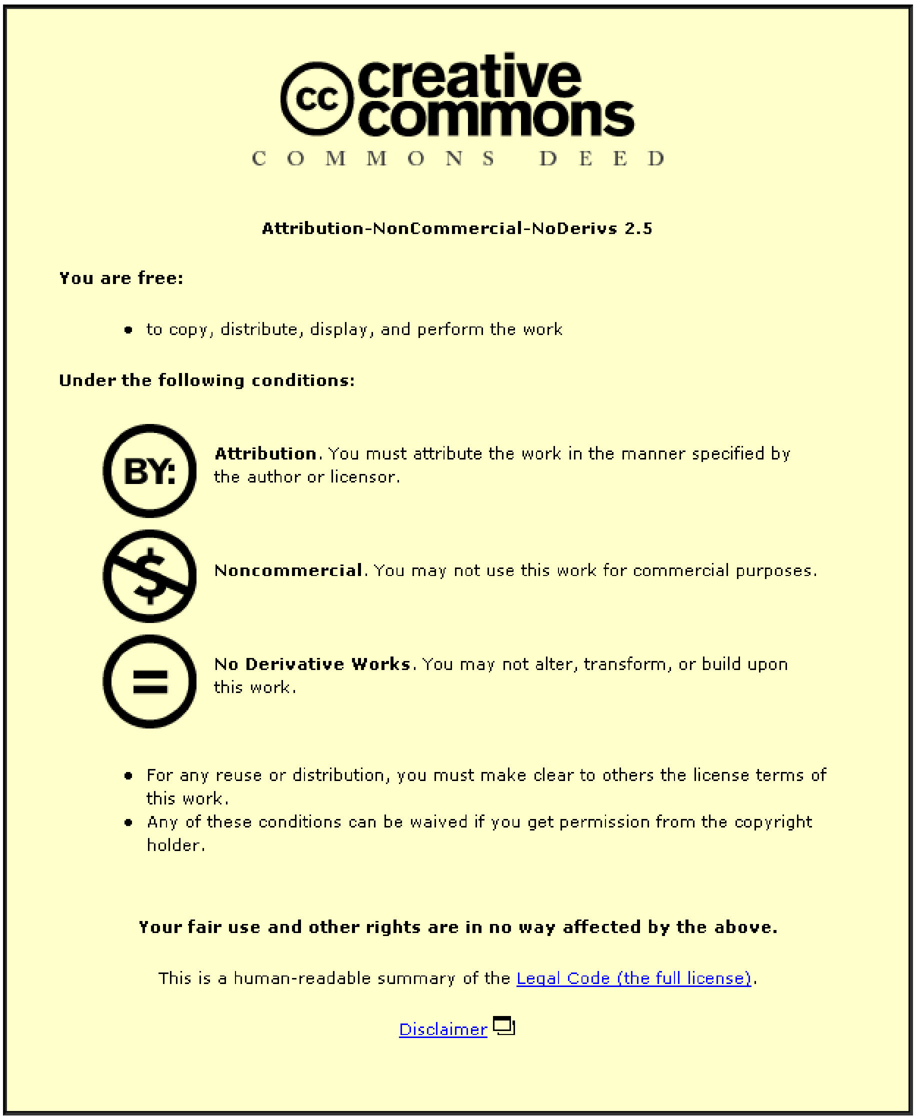

2009 International Conference on Advanced Information Networking and Applications Workshops

Security of Two Recent Constant-Round Password Authenticated Group Key Exchange Schemes

Raphael C.-W. Phan[†]
Electronic & Electrical Engineering
Loughborough University, UK
Email: r.phan@lboro.ac.uk

Abstract

When humans interact with machines in their daily networks, it is important that security of the communications is offered, and where the involved shared secrets used to achieve this are easily remembered by humans. Password-based authenticated group key exchange (PAGKE) schemes allow group users to share a session key based on a human-memorizable password. In this paper, we consider two PAGKE schemes that build on the seminal scheme of Burmester and Desmedt. We show an undetectable online dictionary attack on the first scheme, and exploit the partnering definition to break the key indistinguishability of the second scheme.

Keywords: *Password-authenticated key exchange, group, model, proof, cryptanalysis.*

1. Introduction

Password authenticated key exchange (PAKE) schemes [5], [11], [12] allow two or more parties to share a common secret key to secure subsequent communications. PAKEs are popularly studied because most systems depend on human interactions, as is relevant to the context of present day ubiquitous environments; thus using a password is more practical than a high-entropy secret key. If a human user is assigned a high-entropy secret, he would try at first instance to change it to an easily memorizable password, or be tempted to write it down somewhere or have his web browser cache it. The first known PAKE secure against dictionary attacks is the Encrypted Key Exchange (EKE) by Bellovin and Merritt [5], for 2 parties. Subsequently, PAKEs for groups (3 parties or more) have been proposed, e.g. [2], [9].

Research in group PAKEs can be seen to proceed in two directions: one initiated by Abdalla et al. [2] where each member shares a different password with a server, so called the DPWA setting; while the other initiated by Bresson et al. [9] where group members all share the same password known as the SPWA setting.

Perhaps the most well known scalable group key exchange is the one by Burmester and Desmedt [11], in the SPWA setting. This is well suited for dynamically changing groups since the number of rounds of messages sent between group members is constant, irrespective of the group size. Extensions to this scheme appear in [17], [1]. The recent PAGKE scheme by Kwon et al. at IWSEC 2006 [17] is an adaptation of the Burmester-Desmedt to the SPWA password setting, and is said to be the first provably secure constant-round PAGKE in the standard model (without requiring ideal random oracles). As far as we know, only three constant-round SPWA-type PAGKE schemes secure in the standard model exist, two of which are analysed in this paper: the scheme by Kwon et al. and the scheme by Abdalla and Pointcheval [3].

Our Results. We treat two PAGKE schemes in the SPWA setting: the $\mathcal{PAGKE}$ scheme by Kwon et al. [17] and the scheme by Abdalla and Pointcheval [3]. The benefit of schemes in this setting is that no trusted server S is required.

First, we give an undetectable online dictionary attack against $\mathcal{PAGKE}$. Our attack exploits verifiable messages that are similar to the offline dictionary attacks by [1] against [14], [18]. We then show that due a correctness problem, key indistinguishability of the Abdalla-Pointcheval scheme comes under question. To the best of our knowledge, our results are the first known analysis of both these two schemes of [17], [3]. In recent years, the issue of undetectable online dictionary attacks has been raised [13], [21], [22] because while it is accepted that online dictionary attacks cannot be avoided for password-based schemes, the typical mitigation which is outside the

978-0-7695-3639-2/09 $25.00 © 2009 IEEE
DOI 10.1109/WAINA.2009.42

scope of protocol design, is to limit the number of failed login attempts. However, this mitigation will only work if failed login attempts can be detected in the first place. But for undetectable online dictionary attacks, incorrect password guesses and thus failed login attempts go unnoticed by the legitimate protocol participant being attacked by the adversary. Thus, the adversary's active attacks via Send queries (within the context of the security model; see Section 2.1) cannot be differentiated from honest executions of a legitimate and honest protocol participant, so security against undetectable online dictionary attacks cannot be bound in the same way as detectable online dictionary attacks in terms of Send queries.

We also show that an inconsistency between the partnering definition in the security model and how the group session key is generated, leads to a simple but subtle attack on key indistinguishability that fails to be captured by the scheme's proofs. This is related to showing that a scheme does not meet the correctness requirement [3] of AKE schemes, defining that if two instances of protocol participants are partnered and accepted, then both should hold the same session key.

2. Preliminaries

2.1. Model for Group Key Exchange

For completeness, we describe here the group key exchange (GKE) security model [10], [7], [8].

Protocol participants and execution. Let $\mathcal{U}$ be a non-empty set of protocol players or parties. The adversary, A, controls the communications between all protocol players by interacting with the set of oracles, $\Pi^s_{U_i}$, where Π^s_U is defined as the s^{th} instantiation of a player, $U_i \in \mathcal{U}$, in a specific protocol run. A controls the communication channels via queries to the targeted oracles, as below.

- $\mathsf{Send}(\Pi^s_{U_i}, m)$ query. This models A sending messages to instances of players. A gets back from his query the response which oracle $\Pi^s_{U_i}$ would have generated in processing message m. If oracle $\Pi^s_{U_i}$ has not yet terminated and the execution of protocol leads to accepting, variables SIDS are updated. A query of the form $\mathsf{Send}(\Pi^s_{U_i}, \text{“}start\text{”})$ initiates an execution of this protocol.
- $\mathsf{Reveal}(\Pi^s_{U_i})$ query. Any oracle upon receiving such a query and if it has accepted and so holds session key K, sends this to A.
- $\mathsf{Corrupt}(U)$ query. This query allows the adversary to learn the long-lived key of user U. Under the strong corruption model, internal data of any instances of U executing the protocol are also given to A. Under the weak corruption model, only the long-lived key is given to the adversary.
- $\mathsf{Test}(\Pi^s_{U_i})$ query. This oracle query is only available if $\Pi^s_{U_i}$ is “fresh”. It allows to define the indistinguishability-based notion of security for the key, defined by the following game, denoted $\mathbf{Game}^{\mathsf{GKE}}(A, P)$, between adversary A and oracles $\Pi^s_{U_i}$ involved in executions of protocol P. During the game, A can ask any of the above queries, and may only ask the Test query once. Depending on a randomly chosen bit, $b \in \{0, 1\}$, A is given the actual session key if $b = 1$ or a session key drawn randomly from the session key distribution if $b = 0$. Finally, A outputs a guess b'. Informally, A succeeds if it can guess the bit b with non-negligible advantage $\mathbf{Adv}^{\mathsf{GKE}}_{P,A}$ over randomly guessing, where the advantage is defined as

$$\mathbf{Adv}^{\mathsf{GKE}}_{P,A} = 2\Pr[b' = b] - 1.$$

Note that the first three queries: Send, Reveal, Corrupt are common for any kind model for authenticated key exchange protocols, to model the adversary's ability to attack the protocol. The final query Test is used to define the security of the protocol for which the adversary aims to break, in this case, that of the indistinguishability of the session key.

3. The Constant-Round PAGKE Protocols

Let $U_1, \ldots, U_n$ be the identities in lexical order of n users. Denote by $\mathbb{G}$ a finite cyclic group of order $q \in \mathbb{Z}^*_p$, where p and q are two primes such that $p = 2q+1$, and p a safe prime such that the Decisional Diffie Hellman (DDH) is hard in $\mathbb{G}$. g, g_1 and g_2 denote generators of $\mathbb{G}$ of order q such that the discrete logarithmic relation between g_1 and g_2 is unknown. $||$ denotes concatenation. All arithmetic operations in this paper are performed under the group $\mathbb{G}$.

3.1. PAGKE Schemes in SPWA Setting

The PAGKE scheme designed in the SPWA setting by Kwon et al. [17] at IWSEC 2006 builds on the Burmester-Desmedt scheme in [11]. Since the Burmester-Desmedt scheme is designed without authentication of messages, Kwon et al. suggest to authenticate broadcast messages from group members by masking with a function of the group password. To be precise, this scheme which they called as $\mathcal{PAGKE}$, is described in Fig. 1. Meanwhile, another PAGKE scheme designed in the SPWA setting is by Abdalla and Pointcheval [3] at ASIACRYPT 2006, which builds on the Burmester-Desmedt [11], Gennaro-Lindell [15] and Katz-Ostrovsky-Yung [16] schemes. For compactness, this is specified in Fig. 2.

Some notations are required. (sk_i, vk_i) denotes the signing and verification key pair of U_i, with corresponding $Sign(\cdot)$ and $Ver(\cdot)$ functions. $Enc(pk, l, pw; r)$ denotes tag-based (or labelled) encryption under public key pk and public tag input l and randomness r. UH represents a particular instance of a family of universal hash functions, $ProjKG(\cdot)$ denotes randomized key projection algorithm to generate projective hash keys, and $ProjHash(\cdot)$ denotes the projected hashing algorithm. For exact details, the readers are referred to [3]. For understanding this paper, it is only required to note that $test_i^L$ is not included in T.

4. Attacks on SPWA-based PAGKE Schemes

4.1. $\mathcal{PAGKE}$ Scheme

We show an undetectable online dictionary attack on the Kwon et al. scheme $\mathcal{PAGKE}$ [17].

1) Make a guess pw' of the group password pw. For any x chosen by the adversary, compute $X'_{i-1} = x \cdot g_2^{H(pw'||U_{i-1})}$ and $X'_{i+1} = x \cdot g_2^{H(pw'||U_{i+1})}$.
2) Initiate a session and during Round 1 issue Send queries to send X'_{i-1} and X'_{i+1} to U_i.
3) Thus U_i will compute

$$\begin{aligned}
Y_i &= (\frac{x_{i+1}}{x_{i-1}})^{r_i} \\
&= \left(\frac{\frac{X'_{i+1}}{g_2^{H(pw||U_{i+1})}}}{\frac{X'_{i-1}}{g_2^{H(pw||U_{i-1})}}} \right)^{r_i} \\
&= \left(\frac{\frac{x \cdot g_2^{H(pw'||U_{i+1})}}{g_2^{H(pw||U_{i+1})}}}{\frac{x \cdot g_2^{H(pw'||U_{i-1})}}{g_2^{H(pw||U_{i-1})}}} \right)^{r_i} \\
&= \left(\frac{\frac{g_2^{H(pw'||U_{i+1})}}{g_2^{H(pw||U_{i+1})}}}{\frac{g_2^{H(pw'||U_{i-1})}}{g_2^{H(pw||U_{i-1})}}} \right)^{r_i}
\end{aligned}$$

and broadcast $U_i||2||Y_i$.

4) Check if $Y_i = 1$. Otherwise, repeat from step (1.) by making another guess of pw'.

Note that U_i will not notice that anything is wrong or that an incorrect password was used because the scheme does not explicitly check that passwords used by other group members are correct.

In fact, the adversary can do better by mounting this attack in parallel, i.e. simultaneously making n different password guesses, and launching the attack against all U_i for $i = 1 \ldots n$. Thus, each session allows to verify n password guesses instead of just 1.

4.2. Abdalla-Pointcheval Scheme

Abdalla and Pointcheval presented [3] a constant-round PAGKE scheme based on the group key exchange scheme of Burmester-Desmedt [11] and the 2-party PAKE schemes of Gennaro-Lindell [15] and Katz-Ostrovsky-Yung [16]. The scheme works in broadcast mode, where messages sent from a group member is received by all other members.

Our main observation is the definition of session identifier sid used subsequently in the definition of *partnering* within the security model defined by Abdalla and Pointcheval in [3]. For completeness, we restate the definitions here.

Partnering [3]. *Let the session identifier* sid^i *of a participant instance* U^i *be a function of all the messages sent and received by* U^i *as specified by the group key exchange protocol. Let the partner identifier* pid^i *of a participant instance* U^i *be a set of all participants with whom* U^i *wishes to establish a common secret key. Two instances* U_1^i *and* $U_2^{i'}$ *are said to be partnered if and only if* $\mathsf{pid}_1^i = \mathsf{pid}_2^{i'}$ *and* $\mathsf{sid}_1^i = \mathsf{sid}_2^{i'}$*.*

This is a standard definition of sid and partnering. In fact, the use of sid^i in their scheme specification in Fig. 1 of [3] differs from the standard sid^i definition given in their model, i.e. here the sid^i is given as a function of only some but not all messages sent and received by each group member. Thus, there are in fact two versions of sid^i definitions in [3].

We give a simple attack similar to the attack in Section 3.2 that works on the Abdalla-Pointcheval scheme when partnering is defined as above in their security model.

1) Adversary A launches an $\mathsf{Execute}$ query for a protocol session involving group members $U_1, \ldots, U_n$. This models the adversary eavesdropping on messages exchanged among members during this session.

Round 1:

Each group member U_i chooses a random number $r_i \in_\$ \mathbb{Z}_q^*$, computes $x_i = g_1^{r_i}$ and $X_i = x_i \cdot g_2^{H(pw||U_i)}$. Each U_i broadcasts $U_i||1||X_i$, where 1 represents the broadcast message in the first round.

Round 2:

Each U_i computes x_{i-1} and x_{i+1} using pw and the sender's identities U_{i-1} and U_{i+1}, respectively. Then U_i computes $Y_i = (x_{i+1}/x_{i-1})^{r_i}$ and broadcasts $U_i||2||Y_i$.

Key computation:

Each U_i computes the secret key for F as $k_i = (x_{i-1}^{nr_i}) \cdot Y_i^{n-1} \cdot Y_{i+1}^{n-2} \cdot \ldots Y_{i-2} = g_1^{r_1 r_2 + r_2 r_3 + \cdots + r_n r_1}$, and the session key $sk_i = F_{k_i}(\mathcal{U}||\mathsf{sid}')$, where $\mathcal{U} = (U_1, \ldots, U_n)$, $\mathsf{sid}' = 1||\mathsf{X}||2||\mathsf{Y}$, $\mathsf{X} = (X_1, \ldots, X_n)$, and $\mathsf{Y} = (Y_1, \ldots, Y_n)$.

Figure 1. $\mathcal{PAGKE}$ scheme of IWSEC 2006 [17]

2) A then initiates a new protocol session involving the same group members, this time issuing a Send query in Round 4 to replace broadcast message $(X_i, test_i^L)$ from U_i to U_j ($j \in \{i+1, i-1\}$) with $(X_i, test_i^{\prime L})$ where $test_i^{\prime L}$ is the Round 4 message of the previous session. The rest of the protocol steps proceed normally.
3) A issues a Reveal query to U_j and obtains the session key sk.
4) A issues a Test query to U_i and obtains the test session key sk^*.
5) A checks if $sk^* = sk$ and outputs $b = 1$ if it is. Otherwise it outputs $b = 0$.

Note that based on the standard definition of sid in the Abdalla-Pointcheval model [3] restated above, U_i and U_j will have different sids thus they will not be partnered. Thus, issuing a Reveal query to U_j does not violate the freshness of the instance of U_i, but in fact U_i and U_j will compute the same session key because the only difference between this session and an honest execution is that $test_1^L$ to U_j is replaced, but $test_i^L$ is not used by U_j in verifications nor computations that affect the established session key since it only uses messages from its left and right neighbours; thus it will equal the session key computed by other group members. Thus, changing $test_i^L$ for U_j necessarily translates to U_j not being partnered with other group members although U_j computes the same session key as others. Again, this is an issue related to correctness of the scheme.

In fact, if the definition of sid is not a function of all messages but of messages excluding $test_i^L$, then our attack above no longer works. However, this would be a non-standard definition that differs from existing literature [4], [10].

Interestingly, the earlier scheme by the same authors Abdalla et al. [1] for which security can only be proven in the random oracle and ideal cipher models, resists this attack because there is correctness issue. To be precise, session key computation is a function of all broadcast messages, thus ensuring that partnered instances will compute the same key, and vice versa.

5. Conclusion

Our results cover two of three known PAGKE schemes provably secure in the standard model. The third scheme, by Bohli et al. [6], appears to resist attacks that we have presented. It is also known [3] that the Bohli et al. scheme is more efficient that the Abdalla-Pointcheval scheme. Thus current results gives more preference towards the Bohli et al. scheme than the other two schemes in [17], [3].

The approach we take is one of analyzing security protocols often with accompanying provable security proofs. Such protocols attract analysis attention since they were by right rigorously designed and analyzed within formally defined security models; thus any claim-invalidating attacks potentially reveal insights into subtleties missed by existing proofs or proof techniques. See [20], [23], [24] for other related work in this direction.

Acknowledgements

Part of this work done while the author was with the Laboratoire de sécurité et de cryptographie (LASEC), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. We thank anonymous referees for comments. We thank God for granting us the ability to share secrets [Lk 12:2].

References

[1] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval. Password-Based Group Key Exchange in a Constant Number of Rounds. In *Proc. PKC '06*, LNCS 3958, pp. 427-442, Springer-Verlag, 2006.

[2] M. Abdalla, P.-A. Fouque, and D. Pointcheval. Password-Based Authenticated Key Exchange in the Three-Party Setting. In *Proc. PKC '05*, LNCS 3386, pp. 65-84, Springer-Verlag, 2005.

[3] M. Abdalla, and D. Pointcheval. A Scalable Password-Based Group Key Exchange in the Standard Model. In *Advances in Cryptology - ASIACRYPT '06*, LNCS 4284, pp. 332-347, Springer-Verlag, 2006.

[4] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology - EUROCRYPT '00*, LNCS 1807, pp. 139-155, Springer-Verlag, 2000.

[5] S.M. Bellovin, and M. Merritt. Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks. In *Proc. IEEE S&P '92*, pp. 72-84, IEEE Press, 1992.

[6] J.-M. Bohli, M.I.G. Vasco and R. Steinwandt. Password-Authenticated Constant-Round Group Key Establishment with a Common Reference String. IACR ePRint Archive, 27 June 2006, `http://eprint.iacr.org/2006/214.`

[7] E. Bresson, O. Chevassut and D. Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange - the Dynamic Case. In *Advances in Cryptology - ASIACRYPT '01*, LNCS 2248, pp. 290-309, 2001.

[8] E. Bresson, O. Chevassut and D. Pointcheval. Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. In *Advances in Cryptology - EUROCRYPT '02*, LNCS 2332, pp. 321-336, 2002.

[9] E. Bresson, O. Chevassut, and D. Pointcheval. Group Diffie Hellman Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology - ASIACRYPT '02*, LNCS 2501, pp. 497-514, Springer-Verlag, 2002.

[10] E. Bresson, O. Chevassut, D. Pointcheval and J.-J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In *Proc. ACM CCS '01*, pp. 255-264, 2001.

[11] M. Burmester, and Y. Desmedt. A Secure and Efficient Conference Key Distribution System (Extended Abstract). In *Advances in Cryptology - EUROCRYPT '94*, LNCS 950, pp. 275-286, Springer-Verlag, 1994.

[12] M. Burmester, and Y. Desmedt. A Secure and Scalable Group Key Exchange System. In *Information Processing Letters*, Vol. 94, No. 3, pp. 137-143, 2005.

[13] Y. Ding, and P. Horster. Undetectable On-line Password Guessing Attacks. In *ACM Operating Systems Review*, Vol. 29, No. 4, pp.77-86, 1995.

[14] R. Dutta, and R. Barua. Password-based Encrypted Group Key Agreement. In *International Journal of Network Security*, Vol. 3, No. 1, pp. 30-41, 2006.

[15] R. Gennaro, and Y. Lindell. A Framework for Password-based Authenticated Key Exchange. In *Advances in Cryptology - CRYPTO '03*, LNCS 2656, pp. 524-542, Springer-Verlag, 2003.

[16] J. Katz, R. Ostrovsky, and M. Yung. Efficient Password-Authenticated Key Exchange using Human-Memorable Passwords. In *Advances in Cryptology - EUROCRYPT '01*, LNCS 2045, pp. 475-494, Springer-Verlag, 2001.

[17] J.O. Kwon, I.R. Jeong, and D.H. Lee. Provably-Secure Two-Round Password-Authenticated Group Key Exchange in the Standard Model. In *Proc. IWSEC '06*, LNCS 4266, pp. 322-336, Springer-Verlag, 2006.

[18] S.-M. Lee, J.Y. Hwang, and D.H. Lee. Efficient Password-based Group Key Exchange. In *Proc. TrustBus '04*, LNCS 3184, pp. 191-199, Springer-Verlag, 2004.

[19] J. Nam. Enhancing Security of a Group Key Exchange Protocol for Users with Individual Passwords. IACR ePrint Archive, 2007/166, 2007.

[20] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of an Improved Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) Scheme. In *Proc. ACNS '05*, LNCS 3531, pp. 33-39, Springer-Verlag, 2005.

[21] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of the N-Party Encrypted Diffie-Hellman Key Exchange using Different Passwords. In *Proc. ACNS '06*, LNCS 3989, pp. 226-238, Springer-Verlag, 2006.

[22] R.C.-W. Phan, and B.-M. Goi. Cryptanalysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols. In *Progress in Cryptology - Indocrypt '06*, LNCS 4329, pp. 104-117, Springer-Verlag, 2006.

[23] K. Ouafi, and R.C.-W. Phan. Privacy of Recent RFID Authentication Protocols. In *Proc. ISPEC '08*, LNCS 4991, pp. 263-277, Springer-Verlag, 2008.

[24] K. Ouafi, and R.C.-W. Phan. Traceable Privacy of Recent Provably-Secure RFID Protocols. In *Proc. ACNS '08*, LNCS 5037, pp. 479-489, Springer-Verlag, 2008.

```
Group member $U_i$
$\mathsf{pid}_i = \{U_1, \ldots, U_n\}$
$(sk_i, vk_i) \xleftarrow{\$} \mathsf{SKG}(1^k)$
$l_i = vk_i||U_1|| \ldots ||U_n$
$c_i^R = Enc(pk, l_i, pw; r_i^R)$

Broadcast: $\xrightarrow{l_i, c_i^R}$

$hk_i^L \xleftarrow{\$} HK(pk)$
$phk_i^L \xleftarrow{\$} ProjKG(hk_i^L, l_{i-1}, c_{i-1}^R)$
$c_i^L = Enc(pk, l_i, pw; r_i^L)$

Broadcast: $\xrightarrow{phk_i^L, c_i^L}$

$hk_i, hk_i^R \xleftarrow{\$} HK(pk)$
$phk_i \xleftarrow{\$} ProjKG(hk_i, l_{i+1}, c_{i+1}^L)$
$phk_i^R \xleftarrow{\$} ProjKG(hk_i^R, l_{i+1}, c_{i+1}^L)$
$K_{i+1}^L = ProjHash(phk_{i+1}^L, c_i^R, l_i, pw, r_i^R)$
$K_i^R = Hash(hk_i^R, c_{i+1}^L, l_{i+1}, pw)$
$X_i^R = K_{i+1}^L \cdot K_i^R$
$test_i^R = UH_1(X_i^R)$
$\sigma_i^R = Sign(sk_i, T_i^R)$

Broadcast: $\xrightarrow{phk_i, phk_i^R, test_i^R, \sigma_i^R}$

if $Ver(vk_{i-1}, T_{i-1}^R, \sigma_{i-1}^R) = 0$ then $\mathsf{acc}_i = \mathsf{false}$
$K_i^L = Hash(hk_i^L, c_{i-1}^R, l_{i-1}, pw)$
$K_{i-1}^R = ProjHash(phk_{i-1}^R, c_i^L, l_i, pw, r_i^L)$
$X_i^L = K_i^L \cdot K_{i-1}^R$
if $test_{i-1}^R \neq UH_1(X_i^L)$ then $\mathsf{acc}_i = \mathsf{false}$
$test_i^L = UH_2(X_i^L)$
$K_i = Hash(hk_i, c_{i+1}^L, l_{i+1}, pw)$
$K_{i-1}^R = ProjHash(phk_{i-1}, c_i^L, l_i, pw, r_i^L)$
$X_i = K_i / K_{i-1}$

Broadcast: $\xrightarrow{X_i, test_i^L}$

if $test_{i+1}^L \neq UH_2(X_i^R)$ then $\mathsf{acc}_i = \mathsf{false}$
if $\Pi_{l=1}^n X_l \neq 1$ then $\mathsf{acc}_i = \mathsf{false}$
$T = T_1|| \ldots ||T_n$
$\sigma_i = Sign(sk_i, T)$

Broadcast: $\xrightarrow{\sigma_i}$

for $j = 1, \ldots, i-1, i+1, \ldots, n$
if $Ver(vk_j, T, \sigma_j) = 0$ then $\mathsf{acc}_i = \mathsf{false}$
end for
$MSK = K_i^n \cdot \Pi_{j=1}^{n-1} X_{i+j}^{n-j}$
$SK = UH'(MSK)$
$\mathsf{acc}_i = \mathsf{true}$
$\mathsf{sid}_i = T$
```

Figure 2. Abdalla-Pointcheval scheme of ASIACRYPT 2006 [17], where $T_i^R = U_i||U_{i+1}||c_i^R||c_{i+1}^L||phk_i||phk_i^R||phl_{i+1}^L||test_i^R$ and $T_i = vk_i||U_i||c_i||phk_i||phk_i^L||phk_i^R||X_i||X_i^L$ for $i = 1 \ldots n$.