FORMAL VERIFICATION OF G-PAKE USING CASPER/FDR2
Securing a Group PAKE Protocol Using Casper/FDR2

Mihai-Lica Pura, Victor-Valeriu Patriciu and Ion Bica
Military Informatics and Mathematics Department, Military Technical Academy
81-83 George Cosbuc Boulevard, Bucharest, Romania

Keywords: Formal Verification, Group Password-based Authenticated Key Exchange, Casper, FDR2.

Abstract: Research in security of ad hoc networks consists mainly of classifications and new protocol propositions. But formal verification should also be used in order to be able to prove the properties intended for the protocols. In this paper we present our work in formally verifying the group password-based authenticated key exchange protocol proposed in 2000 by Asokan and Ginzboorg. The proposition is rather old, but in the last years the research community focused only on two-party PAKE protocols, giving very little attention to group PAKE protocols. With the help of Casper and FDR2 we prove that G-PAKE does not accomplish the specifications given by the authors. Based on our results we proposed an improved version that we validated through model checking.

1 INTRODUCTION

Over the last ten years research has generated a large number of password authenticated key exchange (PAKE) protocols. The original protocol in this category dates back from 1992 and was proposed by Bellovin and Merrit with the name of encrypted key exchange (EKE). In few words, the scenario that this protocol addresses is: “two entities who only share a password, and who are communicating over an insecure network, want to authenticate each other and agree on a large session key to be used for protecting their subsequent communications” (Boyko, MacKenzie, Patel, 2001). But the situations in which only two parties need to communicate are unrealistic. In fact, the previous described scenario represent a particular case of the more general one in which several entities communicate over an insecure network and want to create a common secret to be used in exchanging information correctly (Hietalahti, 2001). The protocols that address this last problem and are based on EKE are called group password-based authenticated key agreement protocols.

A lately common application of these protocols is ad hoc networks. Ad hoc networks are infrastructure-less networks that are constructed on the spot in order to respond to a communication need. They are temporary and mobile and so the connections between the nods are usually unreliable. The devices that participate in such networks are often small and portable, which means that their resources (memory, computational power, energy) are constrained. All these characteristics highlight the fact that when developing a protocol targeted on ad hoc network is better not to assume anything about their topology or to assume as little as possible.

Having all these limitation in mind, Asokan and Ginzboorg proposed in 2000 a generic protocol for group password-based authenticated key exchange (G-PAKE) especially for ad hoc networks. In the years that have passed since then, researchers have given very little attention to group PAKE protocols, being more interested in the two-party version. Still a couple of other group PAKE protocols were proposed, but without the special needs of ad hoc networks in mind. For example Yao, Wang, Feng, 2009 proposition that is based on the presence of a central trusted server. These are the reasons for which we consider that this G-PAKE protocol is still important.

G-PAKE’s authors analyze their proposition with regard to the security specifications and targets and conclude that the protocol is safe and it accomplishes the proposed objectives. But, as stated in the paper, this is done without any proof. Arun Kumar Bayya et al. revise this protocol and agree

Pura M., Valeriu Patriciu V. and Bica I. (2010).
FORMAL VERIFICATION OF G-PAKE USING CASPER/FDR2 - Securing a Group PAKE Protocol Using Casper/FDR2.
In *Proceedings of the International Conference on Security and Cryptography*, pages 299-303
DOI: 10.5220/0002980702990303

with the security analysis of Asokan and Ginzboorg. Again, they do not present any proof. Although other password based key agreement protocols have been already formally verified (Tabet, Shin, Kobara, Imai, 2005 and Ota, Kiyomoto, Tanaka, 2009), we did not find such an attempt for this protocol. Our paper presents the formal verification of G-PAKE conducted with Casper and FRD2 tools. Using FDR2 model checker and Casper CSP compiler we found an attack against G-PAKE that prohibits the protocol in assuring its specifications. Based on these results we make a proposition to modify the protocol in order to secure it. By formally verifying our new version we prove that it assures its goals.

The rest of the paper is organized as follows. In section 2 we present G-PAKE protocol as was proposed by the authors. Its intended security properties are described in section 3. Section 4 contains the presentation of the formal verification of G-PAKE and of our version of the protocol. Section 5 contains some conclusions.

2 PROTOCOL PRESENTATION

G-PAKE is based on the basic form of EKE. So we will start by presenting EKE and then we will show how it was extended to multiple parties. In a typical EKE scenario, there are two nodes, i.e. A and B, which share a common weak secret (for example a password). The goals of the protocol are the mutual authentication of A and B based on P, and the agreement on a strong session key K, in such a way that an attacker watching the network traffic will not be able to learn K or to mount an attack on P. Node A owns a key pair formed by an encryption key E_A and a decryption key D_A. During the protocol, node A generates the challenge $challenge_A$ and S_A, node B generates the random number R, the challenge $challenge_B$ and S_B. Considering h is a one-way function and K(msg) is a notation for the result of encrypting the value msg with the key K, the EKE protocol can be summarized as shown below.

```
A->B:A,P(EA)
B->A:P(EA(R))
A->B:R(challengeA,SA)
B->A:R(h(challengeA),challengeB,SB)
A->B:R(challengeB)
```

Figure 1: EKE protocol message exchange.

One can observe that each party generates two nonces: $challenge_A$/$challenge_B$ used by A and B respectively to prove to each other that they know in fact the shared secret P, and S_A/S_B which represent the contribution to the final session key. In order to be easily converted to a contributory multi-party protocol, Asokan and Ginzboorg proposed a modification: the elimination of the challenges and the use of S_A and S_B for both purposes. This leads to a modified version of the protocol that will be used for developing G-PAKE (Figure 2).

```
A->B:A,P(EA)
B->A:P(EA(R,SB))
A->B:R(SA)
A->B:K(SA,h(SA,SB))
B->A:K(SB,h(SA,SB))
```

Figure 2: Asokan and Ginzboorg's EKE protocol version message exchange.

In order to extend this modified version of the protocol to the multi-party case, one of the nodes in the group is elected leader. The leader will initiate the protocol by broadcasting the message in the first step. The rest of the messages will be exchanged in point-to-point communications between the leader and each of the other nodes. Also, the messages from the third and the forth step are sent together.

Considering that the group contains n member nodes, with M_n the elected leader and M_i, with i from 1 to n-1, the rest of the nodes, E and D the encryption/decryption key pair of the leader, P the shared secret and S_i the random share contributed by M_i, the message exchange in G-PAKE is presented in Figure 3.

```
Mn->ALL:Mn,P(E)
Mi->Mn:Mi,P(E(Ri,Si)),  i=1 to n-1
Mn->Mi:Ri({Sj,j=1 to n}), i=1 to n-1
Mi->Mn:Mi, K(S1,H(S1,S2,…,Sn))
```

Figure 3: G-PAKE protocol message exchange.

The authors state that the last step is used for key confirmation. After the third step is completed, each of the nodes, including the leader, can compute the final session key as a function of S_1, S_2, …,S_n.

3 INTENDED SECURITY PROPERTIES

The security properties that the protocol must have derive from its goals: the contributively establishment of a session key K common to all the nodes in the group, based on the password P. In order to achieved these goals, the shared secret P must remain known only to the members of the

group (a), the protocol must be secure against guessing attacks on P (b), leader's encryption key must remain known only by the group members (c), the nonces S_1, S_2, …,S_n must remain secret (d) because they are used to compute the final session key, the leader must authenticate itself to the members in the group (e) and also each member of the group must authenticate to the leader, because only legitimate nodes must be included in a protocol run (f). We give below the formal expression of these properties, using a Casper like syntax. The secrecy properties are expressed through a Secret(A, v,[B]) specification which states that A thinks that v is a secret that can be known to only himself and B.

```
(a) Secret(Mn,P,[M1,...,Mn-1])
(c) Secret(Mn,Si,[M1,...,Mn-1])
(d) Secret(Mn,E,[M1,...,Mn-1])
```

The agreement properties are formalized through Agreement(A,B,[v]) authentication specifications: if responder B completes a protocol run apparently with A, using the data value v, then the same agent A has previously been running the protocol apparently with B, using the same value. And further, each such run of B corresponds to a unique run of A.

```
(e) Agreement(Mn,Mi,[S1,...,Sn])
(f) Agreement(Mi,Mn,[S1,...,Sn])
```

If the guessing attack needs to be verified, it is formally specified by using the reserved word "Guessable".

```
(b) Guessable = P
```

4 FORMAL VERIFICATION

We started the formal verification of G-PAKE with the formal verification of EKE protocol and of the modified version of EKE that the authors proposed in order to be easily transformed into a contributory multi-party protocol. We will not present here the details of these two verifications (the Casper model of these two protocols), because EKE protocol was already verified and proved safe by Lowe. Based on the Casper model provided by Lowe (Lowe, 2001), the modeling of the modified version of EKE is straightforward.

G-PAKE is a multi-party protocol. Casper/FDR2 cannot be used to model and to verify a protocol with an unspecified number of participants. That is why we reduced the protocol to exactly three entities: a leader and other two members. The message exchange between the leader and each of the members is formally the same. If the number of members is higher than two, the only difference will be the corresponding growth of the number of nonces transmitted in steps 3 and 4. There is no reason for which the number of elements in a message will influence its security properties. In conclusion, if the security properties of the protocol will be proved valid on this reduced system, it means that they are valid for a system with any number of members. If the properties will be invalidated by the verification, they wouldn't be valid neither for the general protocol. We conclude by saying that this reduction does not affect the generality of the results.

In Figure 4 the Casper formal specification of original G-PAKE protocol is given. The free variables represent: N – the leader of the group, A and B – the other two members, P – the shared secret, Ra and Rb – the secret keys of the member nodes, sa, sb, sn – the generated nonces, H – a hash function and F – a one-way function for computing the final session key. The F function is defined as "symbolic", which means that the output is not important; the important thing is the fact that its input is the three values generated by the three nodes. For more details about modeling a protocol with Casper, see Lowe, 2001.

```
N->A:N,{PK}{P}
N->B:N,{PK}{P}
A->N:{{Ra,sa}{PK}}{P}
B->N:{{Rb,sb}{PK}}{P}
N->A:{sn,sa,sb}{Ra}
N->B:{sn,sa,sb}{Rb}
A->N:A,{sa,H(sn,sa,sb)}{F(sn,sa,sb)}
B->N:B,{sb,H(sn,sa,sb)}{F(sn,sa,sb)}
```

Figure 4: Casper model of original G-PAKE.

After analyzing the above model, FDR2 concluded that the secrecy specifications (the particularization for this case of the properties presented in section 3) are all valid: P, sa, sb, sn and PK cannot be found by a potential intruder. Also P cannot be guessed. These results confirm the observations given without proof by Asokan and Ginzboorg: the intruder, not knowing and being unable to guess P cannot be part of the protocol, and not knowing sn, sa, sb it cannot generate the final session key.

But the agreement specifications failed. By analyzing the output provided by FDR2 (messages and counterexamples) after they were translated by Casper, we concluded that besides authentication, the contributively nature of the final key is also not achieved. From FDR2 counterexample we saw that the intruder can act like a sort of "man-in-the-middle" between the leader N and the members. Even if the intruder cannot decrypt the messages (we

previously showed that secrecy specifications were verified and that P cannot be guessed) it is capable to eliminate the contribution of a member to the final key. For example, the contribution of member B (sb) to the final key is eliminated, and the contribution of A (sa) to the final key is duplicated. So we concluded that the protocol does not satisfy one of its major purposes: the final key must be created with the contributions of all the members of the group. Also the use of the new key for verification purposes in the final step of the protocol it is not sufficient, as stated by the authors.

We give in Figure 5 the model of our modified version of P-BAKE that successfully accomplishes all the specified security properties:

```
N->A:{N,PK}{P}
N->B:{N,PK}{P}
A->N:{{Ra,sa}{PK}}{P}
B->N:{{Rb,sb}{PK}}{P}
N->A:{sn,sa,sb}{Ra}%tmp1
[decryptable(tmp1,Ra) and
nth(decrypt(tmp1,Ra),2)==sa and
(nth(decrypt(tmp1,Ra),2) !=
nth(decrypt(tmp1,Ra),3))]
N->B:{sn, sa, sb}{Rb}%tmp2
[decryptable(tmp2,Rb) and
nth(decrypt(tmp2,Rb),3)==sb and
(nth(decrypt(tmp2,Rb),2) !=
nth(decrypt(tmp2,Rb),3))]
A->N:{A}{Ra}
B->N:{B}{Rb}
```

Figure 5: Casper model of the modified G-PAKE version.

Our modifications targeted three aspects of the protocol: the transmission of the identities, the verification of the generated values, and the verification of the final session key. In the original version, the leader sends its identity to the members in clear. The members also respond with their identity in clear. We propose to encrypt the identities like all the other elements of the corresponding messages (see messages 1, 2, 7 and 8). Regarding the second aspect, we propose that the member nodes to accept the messages in steps 5 and respectively 6 only if they found their own contribution in the decrypted values and only if their contribution is different from the contribution of the other members (see acceptance condition of the message by the receiver for messages 5 and 6). If the values sn, sa and sb remain secret and if the verifications in step 5 and 6 succeed, we consider that the verification of the computed final key is not necessary; so we proposed a simplified version of the confirmation messages 7 and 8. This represents the third aspect.

By analyzing our model with FDR2 it resulted that all the specified properties are now verified.

5 CONCLUSIONS

In this paper we presented the way we used Casper and FDR2 to check the security properties of G-PAKE protocol. Our verification proved that the secrecy properties are valid, but also revealed that the mutual authentication of the members fails and also that an intruder can perturb the protocol by eliminating the contribution of one or more members to the final key. The elimination of contributions is only possible because the mutual authentication fails. We consider this a very important result, because members' contribution was the main purpose of Asokan and Ginzboorg's proposition. Using this attack on a protocol run for a group with n members, an intruder can eliminate the contributions of maximum n-2 members (except for the leader and for one other member, the contribution of which it will multiply). For that, it will act like a man-in-the-middle between the leader and n-2 members: it will present itself as the leader to n-2 members (by intercepting and resending in its own name the message that the leader board casts in the first step and to the other one member) and as a member to the actual leader (by intercepting and resending in its name the messages of the other one member). This results in a weaker final key, because it will be computed using fewer distinct values (in the worst case only two). So the final key will be much easier to break.

Based on these results and on the original observations of Asokan and Ginzboorg we have proposed a new version of the protocol which achieves mutual authentication of the members and in consequence also resolves the problem with the contributions. We proved both by formal verification. Our version is lighter, because the last message contains fewer elements. We also improved the protocol by moving the verification of the final key (in fact the verification of the values that are used to generate the final key) from the last step to the previous one. This way, in case of an attack, it is not necessary for the nodes to run all the protocol: they will spot the attack and abort without sending the last message.

Because the values used as member's contribution are randomly generated, the acceptance conditions in messages 5 and 6 can be false even in absence of an attack: sa and sb can have the same values because they were generated equal. This is a

drawback of our version of the protocol. Still, because the generation of the s values is random and the generation events for each of the members are independent, the probability that these values are equal is almost zero. In the least favorable case in which the values are equal, the member nodes will abort the corresponding run of the protocol, and the leader will have to start all over again.

Ad hoc networks are, in this moment, a rather theoretical research field. Very few actual implementations exist and even fewer that take into consideration security aspects. So we believe that formal verification of the proposed but not yet used security protocols (as our own) is a very important step towards implementation and standards establishment. Especially because, as we showed in the introduction, this G-PAKE is the only authenticated group key agreement protocol proposition designed especially for ad hoc networks.

REFERENCES

Steven M. Bellovin, Michael Merrit, "Encrypted key exchange: Password-based protocols secure against dictionary attacks", in Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1992.

Victor Boyko, Philip MacKenzie, Sarvar Patel, "Provably Secure Password-Authenticated Key exchange Using Diffie-Hellman", in Proceedings of EUROCRYPT'01, LNCS 1807, Springer-Verlag, 2001.

Maarit Hietalahti, "Key establishment in Ad-hoc Networks", 2001.

N. Asokan, Philip Ginzboorg, "Key Agreement in Ad-hoc Networks", Elsevier Preprint, February 2000.

Arun Kumar Bayya, Siddhartha Gupte, Yogesh Kumar Shukla, Anil Garikapati, "Security in Ad-hoc Networks", Computer Science Department, University of Kentucky.

AbdelilahTabet, Seonghan Shin, Kazukuni Kobara, Hideki Imai, "On Formal Verification Methods for Password-based Protocols: CSP/FDR and AVISPA", in Proceedings of the 4th WSEAS International Conference on Information Security, Communications and Computers, December 2005.

Haruki Ota, Shinsaku Kiyomoto, Toshiaki Tanaka, "Security Verification for Authentication and Key Exchange Protocols", International Journal of Computer Science and Network Security, VOL. 9 No. 3, March 2009.

Gavin Lowe, Philippa Broadfoot, Mei Lin Hui, "Casper A Compiler for the Analysis of Security Protocols User Manual and Tutorial", version 1.5, December 2001

"Failures-Divergence Refinements – FDR2 User Manual", http://www.fsel.com/, June 2005.

Gang Yao, Hongji Wang, Dengguo Feng, "A Group PAKE Protocol Using Different Passwords", in Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009.