A troyan Diffie-Hellman-like protocol based on proof of gullibility

Michela Ceria[1], Alessandro De Piccoli[2], Theo Moriarty[3] and Andrea Visconti[2]

[1] Dipartimento Meccanica Matematica Management, Politecnico di Bari, michela.ceria@poliba.it
[2] Dipartimento di Informatica, Università degli Studi di Milano, {alessandro.depiccoli,andrea.visconti}@unimi.it
[3] Dipartimento di Matematica, Università di Genova, theofmora@gmail.com

Abstract. In the IEEE MILCOM 2018 conference proceedings was published a paper presenting a "lightweight key exchange protocol with provable security". In this *divertissement*, we show that the aforementioned protocol presents a fatal flow that makes the secret key a very simple combination of the public data. Therefore, our main aim is to warn about the intrinsic risks in this protocol and discourage its practical usage, which would cause a leak of information.

Keywords: lightweight cryptography · key exchange · modular arithmetic

1 Introduction

In 2018, the IEEE proceedings of the Military Communications Conference (MILCOM) published the paper [4]. This paper presents a Diffie-Hellmann like key exchange protocol, based on modular arithmetics, which is claimed to be lightweight and with *provable security*. Diffie-Hellmann protocol [2] and its variants [3] are employed as tools for secure key exchange. In particular, the classical protocol and the variant on elliptic curves are used in the TLS/SSL protocol [1]. The variant in [4] is claimed to be particularly suitable for devices with a small amount of resources. A potential user may decide to verify whether the protocol really provides a shared secret, so that the two communicating parties actually get the same result. After this verification, he would trust the paper, believing that the protocol is really secure. Only a paranoid user would verify also the (non-proved) security-providing formula. Well, we are such paranoid users, but also the attacker Eve may be so.

In this brief *divertissement* we show that, verifying the security formula, it turns out that the secret exchanged key is actually a combination of the disclosed data. Therefore, the proposed protocol is not secure, since Eve can retrieve the secret key by eavesdropping the data exchange.

2 The protocol

In this section, we present in details the Diffie-Hellmann-like protocol introduced in [4].

First of all, the paper defines the communicating parties: the *encryptor* E and the *decryptor* D.

After deciding on a public modulus $n \in \mathbb{N}$, $n > 1$, over which all computations are done, both E and D choose their secrets, namely the following integers mod n:

$$E : \{[x_1]_n, [x_2]_n, [x_3]_n, [x_4]_n\}$$

$$D : \{[y_1]_n, [y_2]_n, [y_3]_n, [y_4]_n\},$$

where for each $a \in \mathbb{Z}$, $[a]_n$ means $a \mod n$.

The protocol is sketched in what follows. The values computed by E are indicated as e_i, where the indices $1 \leq i \leq 12$ identify the single variables. Similarly, those computed by D are named d_j, where the indices $1 \leq j \leq 12$ identify the single variables.

Finally, the data which are disclosed between the participants E and D, are indicated with the lower case letter p, followed by an identifying index, namely p_k, $k \in \{1, 2, 3, 4, 5\}$.

All the operations are considered to be done over $\mathbb{Z}_n$.

First of all, E computes

$$[e_1]_n := 2 \cdot ([x_1]_n + [x_2]_n);$$

and dislcloses it under the name of

$$[p_1]_n := [e_1]_n.$$

Once received the value $[p_1]_n$, D can compute the values $[d_1]_n$, $[d_2]_n$, $[d_3]_n$, $[d_4]_n$, $[d_6]_n$, $[d_8]_n$, all depending on the secret data $[y_1]_n, \ldots, [y_4]_n$ and on $[p_1]_n$:

$$\begin{aligned}
[d_1]_n &:= 2 \cdot ([y_1]_n + [y_2]_n) \\
[d_2]_n &:= [y_1]_n + 2 \cdot [y_2]_n \\
[d_3]_n &:= [p_1]_n \cdot [y_1]_n = [e_1]_n \cdot [y_1]_n \\
[d_4]_n &:= [p_1]_n \cdot [d_2]_n = [e_1]_n \cdot [d_2]_n \\
[d_6]_n &:= [d_3]_n - [y_3]_n \\
[d_8]_n &:= [y_3]_n \cdot (2 \cdot [d_4]_n - [d_3]_n) - [d_3]_n \cdot [d_6]_n + [y_4]_n.
\end{aligned}$$

After these computations, D discloses $[d_1]_n, [d_8]_n$, so that E knows

$$\begin{aligned}
[p_2]_n &:= [d_1]_n \\
[p_3]_n &:= [p_1]_n \cdot [p_2]_n \\
[p_4]_n &:= [d_8]_n.
\end{aligned}$$

Basing on the values got from D and on his private data, E can compute:

$$\begin{aligned}
[e_2]_n &:= [x_1]_n + 2 \cdot [x_2]_n \\
[e_3]_n &:= [d_1]_n \cdot [x_1]_n \\
[e_4]_n &:= [d_1]_n \cdot [e_2]_n \\
[e_6]_n &:= [e_3]_n - [x_3]_n \\
[e_8]_n &:= [x_3]_n \cdot (2 \cdot [e_4]_n - [e_3]_n) - [e_3]_n \cdot [e_6]_n + [x_4]_n
\end{aligned}$$

and disclose

$$[p_5]_n := [e_8]_n.$$

Then, D computes the shared secret $[K]_n$ as follows:

$$\begin{aligned}
[d_7]_n &:= (2 \cdot [d_1]_n \cdot [e_1]_n + [d_3]_n + [y_3]_n) \cdot [d_4]_n + [d_3]_n^2 \\
[d_9]_n &:= [d_4]_n \cdot ([d_3]_n + [y_3]_n) \\
[d_{10}]_n &:= [d_7]_n + [d_9]_n \\
[d_{11}]_n &:= [d_{10}]_n + [y_4]_n
\end{aligned}$$

$$[K]_n := [d_{12}]_n := [d_{11}]_n + [e_8]_n \tag{1}$$

and E can independently compute the same secret:

$$\begin{aligned}
[e_7]_n &:= (2 \cdot [d_1]_n \cdot [e_1]_n + [e_3]_n + [x_3]_n) \cdot [e_4]_n + [e_3]_n^2; \\
[e_9]_n &:= [e_4]_n \cdot ([e_3]_n + [x_3]_n) \\
[e_{10}]_n &:= [e_7]_n + [e_9]_n \\
[e_{11}]_n &:= [e_{10}]_n + [x_4]_n
\end{aligned}$$

$$[K]_n := [e_{12}]_n := [e_{11}]_n + [d_8]_n. \tag{2}$$

The key may be also be expressed by means of two values $[q_1]_n, [q_2]_n$, which depend on the private values of both E and D and are claimed not to be computable, neither by E nor by D:

$$[q_1]_n := 2 \cdot ([y_1]_n + [y_2]_n) \cdot [x_2]_n + 2 \cdot ([x_1]_n + [x_2]_n) \cdot [y_2]_n;$$

$$[q_2]_n := [q_1]_n \cdot ([d_3]_n + [y_3]_n + [e_3]_n + [x_3]_n) + 3 \cdot [d_3]_n \cdot [e_3]_n + [x_3]_n \cdot [y_3]_n;$$

$$[K]_n := [e_4]_n^2 + [d_4]_n^2 + 2 \cdot [q_2]_n - 2 \cdot [e_6]_n \cdot [d_6]_n + [x_4]_n + [y_4]_n.$$

3 Correctness and security

The correctness of the protocol (though not proved in [4]) can be easily verified performing the steps of the protocol described in previous section, considering the secret data $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ as indeterminates. Indeed, it turns out that e_{12} and d_{12}, computed as in formulas (1) and (2) respectively, are equal even though the computations are made over $\mathbb{Z}$, so without considering the integer modulus n.

A verification of this kind, made by potential users, can show that the protocol is correct in the sense that the two communicating parties share the same secret.

Then, the paper deals with security, distinguishing among Session-Key Security, Privacy Protection and Known-Key Security. We concentrate on Session-Key Security only, since its failure makes irrelevant this protocol.

The Session-Key Security is based on the following (non-proved) proposition:

Proposition 1. *[4, Prop. 1]*

$$[K]_n - ([p_4]_n + [p_5]_n) = 2([p_3]_n)^2 - [p_3]_n([e_6]_n + [d_6]_n)$$

It is likely that potential users would assume that the above proposition should be correct and the protocol secure. Instead, it is exactly Proposition 1 that presents a *fatal flaw*. Indeed, Eve performs the steps of the protocol to try it on some random numbers and then defines

$$[K']_n := ([p_4]_n + [p_5]_n) + 2([p_3]_n)^2 - [p_3]_n([e_6]_n + [d_6]_n)$$

so if Proposition 1 is correct, then $[K]_n = [K']_n$. On the other hands, when she substracts the value of $[K]_n$ (i.e. $[d_{12}]$ because of the equation (1)) from $[K']_n$, she actually finds

$$[K']_n - [K]_n = -[p_3]_n([e_6]_n + [d_6]_n).$$

This implies that the true value of the key is

$$[K]_n = 2([p_3]_n)^2 + [p_4]_n + [p_5]_n.$$

Therefore, she has verified that the statement of Proposition 1 is wrong.

Even better from her point of view, the public information is enough to get the shared secret and so read all the communication, since $[p_3]_n, [p_4]_n, [p_5]_n$ are public.

Acknowledgment

A.V.: this work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] Dierks T.: "The transport layer security (TLS) protocol version 1.2", RFC 5246, 2008.

[2] Diffie W., Hellman M., "New directions in cryptography", IEEE transactions on Information Theory 22, n.6, 644–654. IEEE, 1976.

[3] Barker, E., Chen, L., Roginsky, A., Vassilev, A. and Davis, R. "NIST Special Publication 800-56A Revision 3. Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography", 2018.

[4] Liu, J.: "A New Lightweight Two-Party Key Agreement Protocol Defined on Underdetermined Systems of Polynomial Equations for Probable Security", *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018.