Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 6236494, 5 pages
http://dx.doi.org/10.1155/2016/6236494

Research Article

lwAKE: A Lightweight Authenticated Key Exchange for Class 0 Devices

Juan Jose Echevarria, Jon Legarda, Janire Larrañaga, and Jonathan Ruiz-de-Garibay

Deusto Institute of Technology, University of Deusto, Avenida de las Universidades 24, Bilbao, 48007 Bizkaia, Spain

Correspondence should be addressed to Juan Jose Echevarria; juanjose.echevarria@deusto.es

Received 14 December 2015; Revised 7 May 2016; Accepted 19 May 2016

Academic Editor: Gianluigi Ferrari

Device-to-Device (D2D) communication enables devices in proximity to establish a wireless direct link. However, these devices may be severely constrained in terms of memory, CPU, and processing resources. Hence, a D2D communication with a constrained device implies new challenges as it does not have the resources required to be secured with standard cryptography. We propose lwAKE for class 0 devices (RFC 7228), which uses one-way cryptographic functions and zero-knowledge proofs to provide mutual authentication and a secure key establishment. We specify the protocol using the High Level Protocol Specification Language and then verify the security properties using the model checkers OFMC and CL-AtSe. The significance of the protocol stands in a key reuse for any successive authentication. Experimental results show that this shortened authentication mode reduces the computational load greatly.

1. Introduction and Related Work

In a Device-to-Device (D2D) communication, authentication plays a very important role. However, devices featuring severe constraints on power, memory, and processing resources imply new challenges for the design of security protocols. RFC 7228 introduces a classification and a terminology for constrained devices which concern data and code size limitations. This classification consists of three classes of constrained devices. We will focus on class 0 devices, which are very constrained sensor-like devices with data/code sizes below 10/100 kB. Thus, authentication in this D2D scenario relies on two parties: the class 0 constrained device (C0) and the device that wants to authenticate and communicate with C0. This device (D0) may be constrained or not.

Most efforts in the design of authentication protocols for C0 devices are primarily focused on RFID tags. Peris-Lopez et al. [1] introduced an ultralightweight RFID authentication protocol that uses index-pseudonyms and a limited collection of binary and rotation operations. However, secrets are updated after a successful protocol round; thus, Yousuf and Potdar [2] blocked the last message to desynchronize the protocol.

Tian et al. [3] introduced another ultralightweight RFID authentication protocol based on novel rotation and permutation operations. The protocol claims that it avoids desynchronization attacks because the reader sends the last message. Ahmadian et al. [4] proved that blocking that last message also breaks the protocol.

Liu and Bailey [5] used random numbers and one-way cryptographic functions. On request, the tag always responds with its constant identifier, which makes the tag traceable. Besides, Naser et al. [6] used that unique identifier to impersonate the tag using the responses of legitimate readers.

Cho et al. [7] proposed a protocol that claims to fulfil the authentication without disclosure of the shared secrets. Safkhani et al. [8] prove the protocol to be prone to desynchronization with a success probability of 1 in only one protocol run. Impersonation attacks are also exposed with a success rate of 1/4 in two runs.

In order to avoid impersonation attacks, Li and Teraoka [9] proposed the use of renewable identifiers. Nevertheless, the fact of using renewable identifiers introduces two issues: increased memory usage and desynchronization attacks against the next valid identifier.

Juels and Weis [10] proposed HB+, which provides a symmetric authentication scheme suited to constrained devices. However, it only provides provable security against passive adversaries. In order to make HB+ resilient against active adversaries, Hammouri and Sunar [11] merged physically unclonable functions (PUF) with the HB+ authentication scheme. A PUF is a physical entity that is embedded in the device, which means this protocol cannot be used in devices that do not have that physical entity built-in.

There are also works that try to fit the public-key cryptography (PKI) as used in most Internet security standards in C0 devices, as in Sample et al. [12]. However, PKI was not designed with energy nor computation constraints in mind. It requires high memory and expensive computations, and C0 devices do not have those resources; for example, Pendl et al. [13] executed a scalar multiplication using the Montgomery powering ladder in 1.6 seconds at a frequency of 6.7 MHz. This execution time is not practical for a C0 device.

Finally, presented works only focus on the authentication. In a D2D communication, we also need to encrypt all messages of the communication session. In this paper, we propose a lightweight authenticated key exchange for C0 devices which establishes the session key and authenticates the devices involved in the key exchange. Furthermore, the protocol conveys a zero-knowledge proof and reduces the number of computations in successive sessions.

The remainder of the paper is structured as follows. We first introduce the proposed authentication protocol for C0 devices. Afterwards, we present the automated analysis of the protocol using two model checkers. Then, we show some energy measurements in our C0 testbed. Finally, the paper ends with a conclusion.

2. Lightweight Authenticated Key Exchange

The proposed protocol provides mutual authentication and key exchange using a limited collection of cryptographic functions and bitwise operations. Furthermore, it presents a second mode that reuses the former session key for any successive authentication. The notation of the protocol is shown in Notation of the Protocol section.

2.1. Initial Authentication. Our protocol keeps two secrets, K_1 and K_2, that are preconfigured in C0 and D0. To produce the session key and the cookies we use a keyed hash function optimized for speed on short messages (SIPHASH), developed by Aumasson and Bernstein [14].

Next we show the steps of this authentication mode:

(1) D0 generates a nonce (nonce_1) and sends a request message with it.

(2) C0 generates a nonce (nonce_2) and computes the session key (K) and its evidence (proof_c). Finally, this information is masked in a one-time cookie. The next set of equations show the key and cookie generation algorithms:

$$K = H\left(K_1, K_2, \text{nonce}_1, \text{nonce}_2\right)$$

$$\begin{aligned} \text{proof}_c &= \left(\text{nonce}_1 \oplus K_{32-63}\right) + \text{nonce}_2 \\ \text{cookie}_c &= H\left(K_1, \text{nonce}_1\right) \oplus \left(\left(\text{proof}_c \ll 32\right) \text{nonce}_2\right). \end{aligned} \tag{1}$$

(3) D0 receives the cookie and validates it. The next set of equations show the cookie extraction:

$$\begin{aligned} \text{nonce}_2 &= \left(\text{cookie}_c \oplus H\left(K_1, \text{nonce}_1\right)\right) \\ &\quad \& \; 0xFFFFFFFF \\ \text{proof}_c &= \left(\text{cookie}_c \oplus H\left(K_1, \text{nonce}_1\right)\right) \gg 32. \end{aligned} \tag{2}$$

(4) D0 recovers nonce_2, computes the session key, and verifies proof_c. If valid, C0 is authenticated and D0 generates a pseudonym identifier for C0 (C_{ID}) to be used on successive authentications. Additionally, it computes its evidence (proof_d) with another part of the key:

$$\text{proof}_d = \left(\text{nonce}_2 \oplus K_{0-31}\right) + \left(\text{nonce}_1 \oplus C_{\text{ID}}\right). \tag{3}$$

(5) Finally, D0 computes the cookie and forwards it to C0:

$$\text{cookie}_d = H\left(K_2, \text{nonce}_2\right) \oplus \left(\left(\text{proof}_d \ll 32\right) \mid C_{\text{ID}}\right). \tag{4}$$

(6) C0 now checks the legitimacy of D0. The next set of equations show the cookie validation algorithm:

$$\begin{aligned} C_{\text{ID}} &= \left(\text{cookie}_d \oplus H\left(K_2, \text{nonce}_2\right)\right) \\ &\quad \& \; 0xFFFFFFFF \\ \text{proof}_d &= \left(\text{cookie}_d \oplus H\left(K_2, \text{nonce}_2\right)\right) \gg 32. \end{aligned} \tag{5}$$

(7) If the evidence is valid, C0 stores its pseudonym and the session key (K). The session key could now be used as the cryptographic key to encrypt the data.

The success of the authentication phase depends on the verification of the session key between C0 and D0. Each one has to check that the other end has come to the same session key. Nevertheless, the session key is never completely sent. In other words, the protocol conveys a zero-knowledge proof and requires knowing more than the session key to break it.

2.2. Successive Authentication. The aim of this mode is to reduce the computing costs in successive connections. Next we show the steps of this authentication mode:

(1) D0 generates a nonce (nonce_a) and calculates two messages (A, B). Then, it sends the messages with its identifier (D_{ID}). The initial z equals 0:

$$\begin{aligned} A &= \left(\text{nonce}_a \oplus K_{z-z+31}\right) \\ x &= \text{nonce}_a \;\&\; 0x3F \\ B &= \left(C_{\text{ID}} \oplus K_{x-x+31}\right) + \text{nonce}_a. \end{aligned} \tag{6}$$

(2) After receiving the messages, C0 gets D0's shared secrets.

(3) Next, C0 infers the value of the nonce (nonce_a) from A and checks the legitimacy of D0 calculating a local version of B. If valid, D0 is authenticated.

(4) Then, C0 generates another nonce (nonce_b), calculates the messages E and F, and sends a response:

$$\begin{aligned} E &= \left(C_{\text{ID}} \oplus \text{nonce}_b\right) + K_{z+x-z+x+31} \\ y &= \text{nonce}_b \ \&\ 0x3F \\ F &= \left(\text{nonce}_a \oplus K_{y-y+31}\right) + \left(\text{nonce}_b \oplus K_{x-x+31}\right). \end{aligned} \tag{7}$$

(5) D0 first infers the nonce (nonce_b) from E and then authenticates C0 calculating a local version of F. If valid, mutual authentication is proved.

(6) Finally, new z is computed only if the connection termination runs properly, which avoids desynchronization:

$$z = \left(\text{nonce}_a \oplus \text{nonce}_b\right) \ \&\ 0x3F. \tag{8}$$

Note that the session key (K) is 64 bits long, and thus the bits selected by x, y, and z will roll over on overflow; for example, if z is 63, the partial session key in message A will be K_{63-30}. Moreover, the fact that message A changes with each iteration, because of z, prevents replay attacks. Successive authentications will keep using this mode unless C0 and D0 distrust each other.

3. Security Validation

Many security protocols do not succeed in their stated goals due to the absence of formal automated analysis. Manually computed validations, even the formal ones, share the limitation of the human processing.

We use AVISPA, a suite of back-end model checkers commonly used for automated validation and verification of cryptographic protocols developed by Armando et al. [15].

From its four back-end model checkers, we use OFMC and CL-AtSe because they support Exclusive-OR properties. CL-AtSe applies constraints on the active intruder knowledge by running the protocol in all possible ways by using redundancy elimination techniques [16]. OFMC uses symbolic techniques and optimizations to perform a bounded analysis for modelling an active intruder in a demand-driven way [17].

The protocol and the assumptions are specified in the High Level Protocol Specification Language (HLPSL). HLPSL is a specification language developed by von Oheimb [18] that uses formal semantics based on Lamport's temporal logic of actions for modelling security-sensitive protocols.

However, HLPSL does not support the arithmetic operators lwAKE uses. Hence, to validate our protocol, we model approximations that consider the security properties that are most important for those operators. We make the modelling assumption that the agents should simply compute a function of the target parameter, the one computed with the unsupported operator, on the basis that an intruder cannot compute the parameter without knowing the secret used in the function. We want to highlight that these are not simplifications, just approximations that consider the security properties of an addition and still render valid the security analysis. Arithmetic addition is basically a bitwise XOR with an AND and a left shift.

We model the message exchange using two agents: C0 and D0. To emulate the actions of an arbitrary adversary, the model uses the Dolev-Yao intruder [19]. In addition to the knowledge of the protocol, it can eavesdrop and generate and intercept messages and use them as an input for its knowledge base and deduction process.

Next, we verify two properties referred to as security properties in IETF documents. First, we evaluate *message authentication*. This security property checks if the received message has been created by a certain device. The aim of this simulation is to validate the mutual authentication of the protocol. This means that C0 and D0 must authenticate each other for the two authentication modes. OFMC and CL-AtSe return a SAFE status with 3 sessions, which means the protocol complies with mutual authentication.

Finally, we evaluate *confidentiality*. This security property states that the protocol parameters are not made available or disclosed to unauthorized devices. In such attack, the adversary tries to impersonate or deceive a legitimate device through the reuse of the information obtained in previous authentications.

As the model only supports single traces, we run several simulation rounds using the previous messages in the attacker's knowledge set to check whether it is possible to reveal a previous session or not.

OFMC results in a SAFE status with 6 bounded sessions and 640 visited nodes with a depth of 8 plies for the initial authentication mode. In the successive authentication mode, the SAFE status is achieved with 6 sessions, 328 visited nodes, and 6 plies. CL-AtSe results in a SAFE status with 6 sessions and 324 analysed states for the initial authentication mode, while the successive mode results in 194 analysed states. This means that the protocol meets the confidentiality property.

To summarize, during the initial authentication, C0 and D0 compute the session key and exchange a zero-knowledge proof of it. If the random number generator is properly designed, an attacker cannot predict it. Conversely, a successive authentication relies on the rotation of the session key and random numbers to protect the confidentiality. Furthermore, the first message of this authentication mode changes on every authentication round, which prevents the reuse of that message.

4. Energy Measurements

This section describes a small energy benchmark of the proposed protocol in our C0 testbed [20]. The testbed uses an 8-bit ATmega640 microcontroller with 64 kB of ROM and 8 kB of RAM. Besides, the current in its active mode is 20 mA (5 V @ 16 MHz).

Table 1: Measurements.

	Mean execution time	Energy	mAh
Initial auth. (1st round)	6300 μs	0.63 mJ	$3.5e^{-5}$
Initial auth. (2nd round)	4400 μs	0.44 mJ	$2.4e^{-5}$
Initial auth. (1st + 2nd)	10700 μs	1.07 mJ	$5.9e^{-5}$
Successive auth.	170 μs	0.017 mJ	$9.4e^{-7}$

Table 1 shows some energy measurements for the different modes of the protocol.

We can see that the cryptographic function takes the most part of the execution time, especially appreciable in the first round of the initial authentication as we compute two cryptographic functions. These results prove that our protocol can greatly reduce the battery drain from reusing the session key in a successive authentication (193.72% difference).

5. Conclusion

This work is a step forward to the design of energy-efficient authentication protocols for C0 devices that need to communicate securely on a relatively frequent basis. Experimental results prove that running the authentication with the key reuse makes our solution more lightweight.

However, we were unable to simulate some skilful attacks with the automated validation tool. Despite being harder to execute, we address the notion that the successive authentication mode is more prone to attacks because of the limited collection of bitwise operations. Future work will focus on a thorough evaluation of the diffusion properties of this mode.

Notation of the Protocol

D_{ID}:	D0's static identifier (32 bits)
C_{ID}:	C0's pseudonym identifier (32 bits)
K_1:	First key
K_2:	Second key
K:	Session key (64 bits)
K_n:	Partial session key with n bit range
nonce:	Random number (32 bits)
proof:	Session key evidence (32 bits)
A, B, E, F:	Messages of the successive authentication mode (32 bits)
cookie:	One-time secret (64 bits)
$H()$:	Keyed hash function (64 bits)
,:	Concatenation
$\oplus$:	Bitwise XOR
|:	Bitwise OR
&:	Bitwise AND
$\ll$:	Left shift
$\gg$:	Right shift
+:	Addition.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work has been supported in part by a Predoctoral Training Fellowship of the Department for Education, Language Policy and Culture of the Basque Government.

References

[1] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda, "LMAP: a real lightweight mutual authentication protocol for low-cost RFID tags," in *Proceedings of the Workshop on RFID Security (RFIDSEC '06)*, p. 6, Graz, Austria, 2006.

[2] Y. Yousuf and V. Potdar, "A survey of RFID authentication protocols," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications-Workshops (AINAW '08)*, pp. 1346–1350, Okinawa, Japan, March 2008.

[3] Y. Tian, G. Chen, and J. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Communications Letters*, vol. 16, no. 5, pp. 702–705, 2012.

[4] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, "Desynchronization attack on RAPP ultralightweight authentication protocol," *Information Processing Letters*, vol. 113, no. 7, pp. 205–209, 2013.

[5] A. X. Liu and L. A. Bailey, "PAP: a privacy and authentication protocol for passive RFID tags," *Computer Communications*, vol. 32, no. 7–10, pp. 1194–1199, 2009.

[6] M. Naser, P. Peris-Lopez, R. Budiarto, and B. R. Álvarez, "Short Communication: a note on the security of PAP," *Computer Communications*, vol. 34, no. 18, pp. 2248–2249, 2011.

[7] J.-S. Cho, S.-S. Yeo, and S. K. Kim, "Securing against brute-force attack: a hash-based RFID mutual authentication protocol using a secret value," *Computer Communications*, vol. 34, no. 3, pp. 391–397, 2011, Special Issue of Computer Communications on Information and Future Communication Security.

[8] M. Safkhani, P. Peris-Lopez, J. C. Hernandez-Castro, and N. Bagheri, "Cryptanalysis of the Cho et al. protocol: a hash-based RFID tag mutual authentication protocol," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 571–577, 2014.

[9] Y. Li and F. Teraoka, "Privacy protection for low-cost RFID tags in IoT systems," in *Proceedings of the 7th International Conference on Future Internet Technologies (CFI '12)*, pp. 60–65, ACM, Seoul, South Korea, September 2012.

[10] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology—CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14–18, 2005. Proceedings*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 293–308, Springer, Berlin, Germany, 2005.

[11] G. Hammouri and B. Sunar, "PUF-HB: a tamper-resilient HB based authentication protocol," in *Applied Cryptography and Network Security*, S. M. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds., vol. 5037 of *Lecture Notes in Computer Science*, pp. 346–365, Springer, Berlin, Germany, 2008.

[12] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 11, pp. 2608–2615, 2008.

[13] C. Pendl, M. Pelnar, and M. Hutter, "Elliptic curve cryptography on the WISP UHF RFID Tag," in *RFID. Security and Privacy*, A. Juels and C. Paar, Eds., vol. 7055 of *Lecture Notes in Computer Science*, pp. 32–47, Springer, Berlin, Germany, 2012.

[14] J.-P. Aumasson and D. J. Bernstein, "SipHash: a fast short-input PRF," in *Progress in Cryptology—INDOCRYPT 2012: 13th International Conference on Cryptology in India, Kolkata, India, December 9–12, 2012. Proceedings*, vol. 7668 of *Lecture Notes in Computer Scienc*, pp. 489–508, Springer, Berlin, Germany, 2012.

[15] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proceedings of the 17th International Conference on Computer Aided Verification (CAV '05)*, vol. 3576 of *Lecture Notes in Computer Science*, pp. 281–285, Springer, July 2005.

[16] M. Turuani, "The CL-atse protocol analyser," in *Term Rewriting and Applications: 17th International Conference, RTA 2006 Seattle, WA, USA, August 12–14, 2006 Proceedings*, vol. 4098 of *Lecture Notes in Computer Science*, pp. 277–286, Springer, Seattle, Wash, USA, 2006.

[17] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: a symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, no. 3, pp. 181–208, 2005.

[18] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of the APPSEM II Workshop*, September 2005.

[19] I. Cervesato, "The Dolev-Yao intruder is the most powerful attacker," in *Proceedings of the 16th Annual Symposium on Logic in Computer Science (LICS '01)*, pp. 16–19, IEEE Computer Society Press. Short, 2001.

[20] J. J. Echevarria, J. Ruiz-de-Garibay, J. Legarda, M. Álvarez, A. Ayerbe, and J. I. Vazquez, "WebTag: web browsing into sensor tags over NFC," *Sensors*, vol. 12, no. 7, pp. 8675–8690, 2012.