Session-HB: Improving the Security of HB$^+$ with a Session Key Exchange

Ahmad Khoureich Ka(✉)

Department of Computer Science,
Alioune Diop University of Bambey, Bambey, Senegal
ahmadkhoureich.ka@uadb.edu.sn

Abstract. The HB$^+$ protocol, designed by Juels and Weis to mitigate forgery and counterfeiting risks on RFID tags, is well suited for those resource-constrained devices. The protocol comes in response to the search for a solution to improve the security of the HB protocol published in 2001 by Hopper and Blum that was not resistant to active attacks. However, Gilbert *et al.* showed that HB$^+$ cannot resist against a simple man-in-the-middle attack. In this paper, we propose to run a lightweight session key exchange as a pre-protocol to establish the tag and reader secrets for HB$^+$. The resulting protocol denoted Session-HB is provably resistant to man-in-the-middle attacks.

Keywords: HB$^+$ · RFID tags · Authentication · LPN
Session key exchange

1 Introduction

The rapid progress we see today in the use of the RFID chips is due to its advantages over barcodes (timeliness in data collection, no need of human involvement, read/write for tags, etc.). RFID tags are used for animal tracking, anti-theft for merchandise in stores, payment and access control. Some of these uses require security, especially authentication. Since the tag can be forged, the design of well-suited authentication protocols, which do not leak sensitive information, is of great need. Well-suited protocols because RFID tags are resource constrained devises, they have no computational power and storage for standard cryptographic tools (e.g. RSA, AES, hash functions. etc.). This has motivated Hopper and Blum to invent the HB protocol [12], a lightweight authentication protocol for low cost RFID tags that has inspired many researchers to propose HB-like protocols. The HB protocol is only resistant to passive adversary but falls in front of active ones. Its resistance to passive attacks lies on the Learning Parity with Noise (LPN) known to be a hard problem [2–4,12,21]. To strengthen HB, Juels and Weis introduce the HB$^+$ protocol [13], which is secure against passive and active attacks [13,15] but not against man-in-the-middle ones

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
C. M. F. Kebe et al. (Eds.): InterSol 2017/CNRIA 2017, LNICST 204, pp. 202–212, 2018.
https://doi.org/10.1007/978-3-319-72965-7_19

e.g. GRS attack [9]. Since that time many researchers have published protocols [5–7,10,16,18] they claim resistant to man-in-the-middle attacks but many of them have weaknesses [8,11,19].

In this paper, we propose Session-HB a new protocol that follows the same framework as hHB introduced by Ka [14]. hHB is a two stages protocol; in the first stage the reader sends a session key to the tag and in the second stage the reader do r HB$^+$ rounds to authenticate the tag. Although hHB has explicit security proofs against man-in-the-middle attacks, its transmission cost is unacceptably high for resource-constrained devices [1]. This drawback of hHB is mainly due to the transmission of the shared secrets by the reader. Using a pre-protocol to renew the tag and reader secrets is not a new idea. We find it in the work of Bringer et al. [6] named HB^{++}. This latter protocol is also a tentative to fix the shortcoming of HB$^+$ that is its weakness against the GRS attack [9]. But HB^{++} doesn't keep its promise to resist to man-in-the-middle attacks [11]. The proposal Session-HB improves the transmission cost of the secrets and at the same time is provably resistant to man-in-the-middle attacks.

This paper is organized as follows: in Sect. 2, we briefly present the LPN problem and describe the HB$^+$ protocol and its weakness. Section 3 exposes our proposal Session-HB, its security arguments and parameter values. Finally, Sect. 5 gives the conclusion.

2 The HB$^+$ Protocol

At Crypto 2005 Juels and Weis presented HB$^+$ an improvement of the HB protocol [12] which exploits the hardness of the Learning from Parity with Noise (LPN) problem. The HB$^+$ protocol is secure against passive and active attacks.

2.1 LPN Problem

The LPN is the problem of finding the k-bit string x from the following system of noisy equations.

$$\begin{cases} a_0 \cdot x \ = z_0 \oplus \nu_0 \\ \cdots \\ a_n \cdot x = z_n \oplus \nu_n \end{cases}$$

where $a_i \leftarrow \{0,1\}^k$, $z_i = a_i \cdot x$ and $\nu_i \sim \mathsf{Ber}_\varepsilon$ the Bernoulli distribution with parameter $\varepsilon \in]0, 1/2[$, (i.e. if $\nu \leftarrow \mathsf{Ber}_\varepsilon$ then $\Pr[\nu = 1] = \varepsilon$ and $\Pr[\nu = 0] = 1-\varepsilon$).

More formally, let $A_{x,\varepsilon}$ be the distribution defined by:

$$\{a \leftarrow \{0,1\}^k; \nu \leftarrow \mathsf{Ber}_\varepsilon : (a, \langle x, a\rangle \oplus \nu)\}$$

The LPN problem is to distinguish oracle access to $A_{x,\varepsilon}$ from oracle access to the uniform distribution on $(k+1)$-bit strings. The LPN is known to be a hard problem [2,3,21] and the best known algorithms for solving it have running time of $2^{\Theta(k/\log k)}$ [3].

2.2 HB$^+$ Design and Weakness

HB$^+$ is a lightweight protocol with a very simple design, see Fig. 1. Its resistance to active attacks comes from the introduction of a random blinding factor b. The reader and the tag share two secrets $x \in \{0,1\}^{k_1}$ and $y \in \{0,1\}^{k_2}$. A round of HB$^+$ consists of the following steps:

1. The tag randomly selects a blinding factor $b \leftarrow \{0,1\}^{k_2}$ and sends it to the reader.
2. The reader responds with a randomly selected challenge vector $a \leftarrow \{0,1\}^{k_1}$.
3. The tag selects ν according to Ber_ε then computes and sends to the reader the bit $z = a \cdot x \oplus b \cdot y \oplus \nu$.

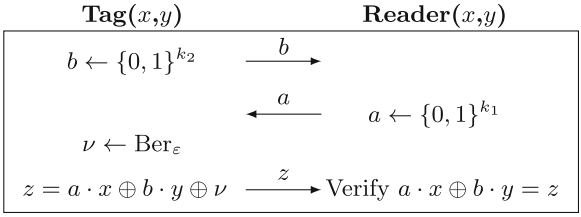

Fig. 1. A round of the HB$^+$ protocol.

The entire authentication process consists of executing r times the HB$^+$ round. The reader recognizes the outcome *yes* or *no* (Verify $a \cdot x \oplus b \cdot y = z$) of each round. If the number of *no* does not exceed a threshold u, the tag is authenticated. One consequence of the probabilistic nature of the authentication is that a honest tag can be rejected by a honest reader (False Rejection) or a counterfeit tag be accepted (False Acceptance). Fortunately, *false rejection* and *false acceptance* happen with negligible probabilities in k_1 (because $r = r(k_1)$):

$$P_{FR} = \sum_{i=\mathsf{u}+1}^{r} \binom{r}{i} \varepsilon^i (1-\varepsilon)^{r-i}, \qquad P_{FA} = \frac{1}{2^r} \sum_{i=0}^{u} \binom{r}{i}$$

The main weakness of HB$^+$ is that it succumbs to man-in-the-middle attacks. A simple man-in-the-middle attack named GRS attack [9] has been successfully mounted against HB$^+$. The GRS attack consists of adding a perturbation e_i (the vector with all 0s but 1 at position i) to the challenge vector a and observe the result of the authentication process of a honest tag. This perturbation is effective if $e_i \cdot x = 1$. Thus if the authentication succeeds with a probability greater than P_{FA}, it means that the bit at the position i of x is 0 otherwise it is 1. The GRS attack is simple and has motivated many researchers to propose solutions for the HB$^+$ protocol [5–7,16,18] but many of them show weaknesses in their design [8,11,19].

3 The Proposal Session-HB

Session-HB follows the idea developed by hHB [14] and HB^{++} [6] that is to use a pre-protocol to renew the tag and reader secrets in the HB^{+} protocol. Session-HB introduces a lighter session key exchange than the one of hHB and unlike HB^{++} is resistant to man-in-the-middle attacks.

3.1 First Stage of Session-HB: A Lightweight Session Key Exchange

The lightweight session key exchange protocol we introduce here is intended to constitute the first stage of Session-HB. The tag and the reader share two k-bit secrets s_1 and s_2. The following steps describe the protocol (see Fig. 2 for a graphical representation):

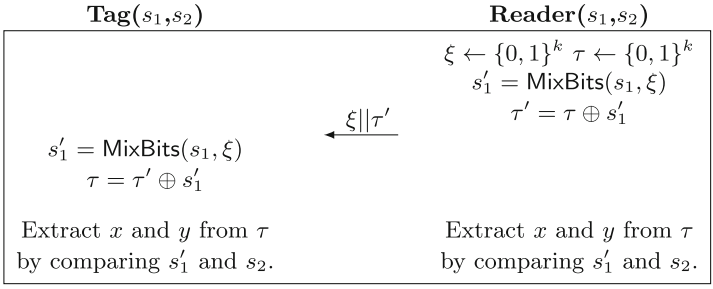

Fig. 2. The first stage of Session-HB for the establishment of session keys x and y used in the second stage.

1. The reader selects ξ and τ randomly from $\{0,1\}^k$, computes $s'_1 = \mathsf{MixBits}(s_1, \xi)$ and $\tau' = \tau \oplus s'_1$. MixBits is a mixing function that is used to randomize the positions where s'_1 and s_2 have the same bits thus making the extraction of the secrets x and y at the final step random. The mixing function also helps to consider s'_1 as a one-time pad even if it's not a perfect one. After that the reader concatenates ξ and τ' and transmits the result to the tag.
2. Upon receiving $\xi||\tau'$, the tag computes $s'_1 = \mathsf{MixBits}(s_1, \xi)$ and retrieve τ.
3. The reader compares s'_1 and s_2 and extracts the session keys x and y from τ. The extraction is done as follow: if at some position i, s'_1 and s_2 have the same bit, the bit of τ at that position belongs to x otherwise it belongs to y. For example if $s'_1 = 00101000$, $s_2 = 01110101$ and $\tau = 10010001$ then $x = 100$ and $y = 01001$. The tag do the same as the reader to obtain the keys x and y. Note that the size of x and y are around $k/2$ as stated in the following theorem.

Theorem 1. *Let s_1 and s_2 be two binary strings of length n as in our lightweight session key exchange. If the binary string $\xi||\tau'$ sent by the reader to the tag in the second step of the lightweight session key exchange is not modified by an active attacker then the length l of the extracted session key x satisfies $l = \Theta(n/2)$.*

Proof (Sketch of the proof). Let $s'_1 = a_1, \ldots, a_n$ and $s_2 = b_1, \ldots, b_n$. Since s_1 and s_2 are randomly and independently selected from $\{0,1\}^k$, s'_1 and s_2 are sequences of independent and identically distributed random variables of expected values $1/2$ and variances $1/4$. Let

$$X_i = \begin{cases} 0 & \text{if } a_i \neq b_i \\ 1 & \text{if } a_i = b_i. \end{cases}$$

X_i is a random variable with expected value $1/2$ and variance $1/4$. The size of the extracted session key x is equal to the sum $X_1 + X_2 + \ldots + X_n$ and by the law of large numbers, for any $\varepsilon > 0$, $Pr(|\frac{l}{n} - \frac{1}{2}| < \varepsilon) \to 1$ as $n \to \infty$. This means that for large n the size of x is around $n/2$.

3.2 Second Stage of Session-HB

The secrets x and y are obtained from the first stage. Their lengths are not fixed ($|x| = \Theta(k/2)$ and $|y| = \Theta(k/2)$ where $k = |s_1| = |s_2|$). Therefore the second stage of Session-HB is identical to the HB$^+$ protocol with some little adjustments (see Fig. 3 for a graphical representation).

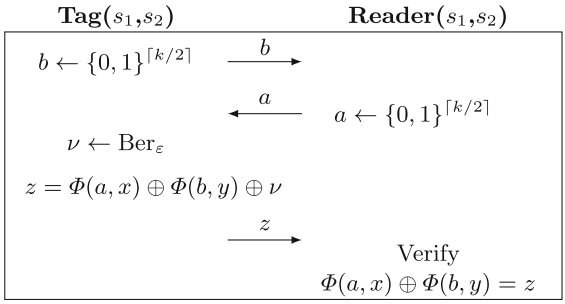

Fig. 3. A single round of the second stage of the Session-HB authentication protocol. It is executed r times for the authentication of the tag.

The steps of one round of the second stage of the Session-HB are the following:

1. The tag randomly selects a blinding factor $b \leftarrow \{0,1\}^{\lceil k/2 \rceil}$ and sends it to the reader, $\lceil k/2 \rceil$ is the ceil of $k/2$.
2. The reader responds with a randomly selected challenge vector $a \leftarrow \{0,1\}^{\lceil k/2 \rceil}$.
3. Instead of computing $a \cdot x \oplus b \cdot y$ as in the HB$^+$ protocol, the tag computes $\Phi(a,x) \oplus \Phi(b,y)$ where Φ is a very simple and lightweight function, see Algorithm 1. So the tag sends to the reader the bit $z = \Phi(a,x) \oplus \Phi(b,y) \oplus \nu$.
4. The reader accepts the round if $z = \Phi(a,x) \oplus \Phi(b,y)$.

The second stage of Session-HB consists of r executions of the above steps.

Algorithm 1. Function Φ

function $\Phi(u, v)$
  $n = |u|$ ▷ $u = u_1 \dots u_n$
  $m = |v|$ ▷ $v = v_1 \dots v_m$
  if $n < m$ **then**
    $i = \lceil \frac{m-n}{2} \rceil + 1, \quad j = i + n - 1$ ▷ $\lceil \frac{m-n}{2} \rceil$: ceil of $\frac{m-n}{2}$
    $v' = v_i \cdots v_j$.
    return $u \cdot v'$
  end if
  if $n >= m$ **then**
    $i = \lceil \frac{n-m}{2} \rceil + 1, \quad j = i + m - 1$
    $u' = u_i \cdots u_j$.
    return $u' \cdot v$
  end if
end function

3.3 Parameter Values for Session-HB

For the mixing function in the first stage of Session-HB, we opted for the one proposed in [20] which is extremely lightweight. It uses only bitwise right shift and additions as shown below.

The MixBits function

```
Z = MixBits(X,Y)
Z = X;
for(i=0; i<32; i++) {
   Z = (Z>>1) + Z + Z + Y ;
}
```

We set the size of the pre-shared keys s_1 and s_2 to 128 bits ($|s_1| = |s_2| = k = 128$ bits), hence the size of the secrets in the second stage will be around 64 ($|x| = |y| = \Theta(64)$). These values do not follow the recommendations of [17] but we think it does not weaken the security of the protocol since x and y are session keys. The false acceptance and the false rejection rates are respectively functions of $(r, \mathtt{u})$ and $(r, \mathtt{u}, \varepsilon)$. We define $r = 1164$, $\mathtt{u} = 0.348 \times r$ and $\varepsilon = 0.25$ and get $P_{FA} = 2^{-80}$ and $P_{FR} = 2^{-40}$.

Table 1. Parameters, storage and transmission costs of some protocols

Protocol	Parameters	Storage costs	Transmission costs
HB$^+$ [13]	$\varepsilon = 0.25; k_x = 80; k_y = 512; r = 1164$	592	690252
HB^{++} [6]	$\varepsilon = 0.25;\ k = 768;\ r = 731$	768	118582
hHB [14]	$\varepsilon = 0.25;\ k_s = 256;\ k_y = 512;\ r = 1164$	768	752703
GHB$^\#$ [22]	$\varepsilon = 0.25;\ k_X = 80;\ k_Y = 512;\ m = 1164$	689088	1756
Session-HB	$\varepsilon = 0.25; k_1 = 128; k_2 = 128; r = 1164$	256	150412

With these settings the transmission cost of Session-HB is significantly lower than that of [14] (see Table 1) and one of the lowest in the family of HB-like protocols [1].

4 Security Arguments

In this section, we prove that Session-HB is secure against active and man-in-the-middle attacks.

4.1 Security Definitions

Active attacks. Such attacks are performed in two stages: a learning phase in which the adversary interacts with a honest tag a polynomial number of times hoping a leak of information about the secrets, and a verification phase in which the adversary tries to authenticate to the reader.

Man-in-the-middle attacks. These are the most powerful attacks against an authentication protocol. The adversary can tamper with messages exchanged between the tag and the reader in a polynomial number of instances of the protocol. Then, he can analyse their effect on the reader's decision (accepting or rejecting the tag) in order to gain information about the secrets. Finally, with the information supposedly obtained on the secrets, the adversary tries to authenticate to the reader.

4.2 Security of Session-HB Against Active Attacks

Theorem 2. *If HB$^+$ with parameters $\varepsilon \in]0, \frac{1}{2}[$, $r = r(k)$ and $\mathsf{u} > \varepsilon \cdot r$ is secure against active attacks then Session-HB with the same settings of parameters is secure against active attacks.*

Proof. In this proof we reduce HB$^+$ to Session-HB. That is we prove that if there is $\mathcal{A}$ a probabilistic polynomial-time adversary that can break Session-HB by an active attack, then we can construct $\mathcal{A}'$ a probabilistic polynomial-time algorithm that can break HB$^+$ by the same type of attack.

Now let's consider $\mathcal{A}$ be a probabilistic polynomial-time adversary interacting with the tag in at most q executions of the Session-HB protocol. Suppose $\mathcal{A}$ can break Session-HB by an active attack with success probability at least ε. We use $\mathcal{A}$ to construct a probabilistic polynomial-time algorithm $\mathcal{A}'$ that performs an active attack on HB$^+$. Let ε' be the success probability of $\mathcal{A}'$. For the learning phase of the attack, $\mathcal{A}'$ uses its interactions with a honest HB$^+$ tag to emulate for $\mathcal{A}$ a Session-HB tag for q times as follow:

First, $\mathcal{A}'$ receives from $\mathcal{A}$ a bit string representing the concatenation $\xi||\tau'$ according to the first stage of the Session-HB Protocol. Second, $\mathcal{A}$ and $\mathcal{A}'$ repeat the following three steps r times (the number of rounds of the Session-HB protocol).

1. $\mathcal{A}'$ responds to $\mathcal{A}$ with the blinding vector b received from the HB$^+$ tag,
2. $\mathcal{A}$ sends a challenge vector a to $\mathcal{A}'$ which in turn forwards it to the HB$^+$ tag,
3. $\mathcal{A}'$ sends the response z of the HB$^+$ tag to $\mathcal{A}$.

For the verification phase of the attack, $\mathcal{A}'$ simulates a Session-HB reader only once for $\mathcal{A}$. First, $\mathcal{A}'$ sends a bit string of length $|\xi||\tau'|$ according to the first stage of the Session-HB protocol. Second, $\mathcal{A}'$ performs with $\mathcal{A}$ the following steps r times:

1. $\mathcal{A}'$ receives from $\mathcal{A}$ a blinding vector b and forwards it to the HB$^+$ reader,
2. $\mathcal{A}'$ receives a challenge vector a from the HB$^+$ reader and forwards it to $\mathcal{A}$,
3. $\mathcal{A}'$ receives from $\mathcal{A}$ a response z and forwards it to the HB$^+$ reader.

Taking into account the way that $\mathcal{A}'$ uses $\mathcal{A}$, one can see that the latter adversary cannot distinguish if he performs an active attack on Session-HB or he is executed as a subroutine by $\mathcal{A}'$. Thus the success probability of $\mathcal{A}$ attacking Session-HB equals the success probability of $\mathcal{A}'$ attacking HB$^+$. That is $\varepsilon = \varepsilon'$. Since ε' is negligible because HB$^+$ is secure against active attacks, then ε is negligible. This concludes the proof.

4.3 Security of Session-HB Against MITM Attacks on the First Stage of the Protocol

Here we prove the security of Session-HB when an adversary $\mathcal{A}$ performs a man-in-the-middle attack on its first stage. The first stage of Session-HB is a lightweight session key exchange, see Fig. 2. We define the following game:

Setup: The tag and the reader have the secrets s_1 and s_2.
Attack: $\mathcal{A}$ has access to a legitimate tag and a legitimate reader and interacts with them q times where each interaction is as follow:
$\mathcal{A}$ receives a commitment $c = \xi||\tau'$ from the reader and sends $c \oplus c'$ to the tag. After that, $\mathcal{A}$ leaves the second stage of Session-HB to proceed normally. The reader outputs *accept* if at least for $r - \mathsf{u}$ rounds of the second stage $\Phi(a, x) \oplus \Phi(b, y) = z$ (see Fig. 3).

Winning Condition: $\mathcal{A}$ wins the game if he makes the reader outputs *accept* for an interaction where $c' \neq 0$.

We now show that an adversary has a negligible success probability of winning the above game, which means that an adversary cannot gain information about the secrets s_1 and s_2.

Theorem 3. *If the adversary cannot win the above game with non negligible probability then he cannot succeed in breaking Session-HB with a man-in-the-middle attack on its first stage.*

Proof. Clearly, if an adversary modifies the concatenation $\xi||\tau'$ sent by the reader to the tag in the first stage of Session-HB, the extracted secrets x and y will be different on both sides (tag and reader). Thus in the second stage (which is similar to the HB$^+$ protocol), from the point of view of the reader, the tag is

sending random responses to challenges. Therefore the probability of winning the game corresponds to the probability of false accept which is negligible. This means the adversary gains no information on the long-lived secrets s_1 and s_2. The theorem follows.

4.4 Security of Session-HB Against MITM Attacks on the Second Stage of the Protocol

Here we make an heuristic analysis of the security of Session-HB when a man-in-the-middle attack is mounted on its second stage.

The second stage of Session-HB is similar to HB$^+$. The view of the adversary when he perturbs q instances of the protocol by adding α, β and γ respectively to a, b and z is close the view of an adversary doing the same against HB$^\#$ (an HB-like protocol introduced by Gilbert *et al.* [10]). Because for Session-HB, these interactions lead to equations involving $q \cdot r$ challenge pairs (a, b), q secret pairs (x, y) and q reader's decisions (r is the number of rounds) while for HB$^\#$ it is q challenge pairs (a, b), r secret pairs (x, y) (in the form of a pair of matrices (X, Y) of r columns each) and q reader's decisions (*accept* or *reject*). The single successful attack [19] we know against HB$^\#$ perturbs the protocol by superimposing α, β and γ obtained from eavesdropping a previous instance of the protocol to other instances. This attacks works on HB$^\#$ because the eavesdropped instance uses the same secrets X and Y. Therefore we believe that, it cannot works on Session-HB because each instance of the protocol uses a different secret pair (x, y). Also, whatever the means used to find the session secrets x or y or both, it is highly unlikely to compute the long-lived secrets s_1 and s_2 from x and y because they are not directly related.

5 Conclusion

We have presented here a new protocol named Session-HB. It follows an idea, implemented in the protocols HB^{++} and hHB, which consists of renewing the HB$^+$ secrets at each authentication process. This was done by introducing a new lightweight session key exchange between the tag and the reader. Contrary to hHB, Session-HB has a significant lower transmission cost which is one of the lowest in the family of HB-like protocols. Unlike HB^{++}, Session-HB is provably secure against man-in-the-middle attacks.

References

1. Armknecht, F., Hamann, M., Mikhalev, V.: Lightweight authentication protocols on ultra-constrained RFIDs - myths and facts. In: Saxena, N., Sadeghi, A.-R. (eds.) RFIDSec 2014. LNCS, vol. 8651, pp. 1–18. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13066-8_1
2. Blum, A., Furst, M., Kearns, M., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_24

3. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM (JACM) **50**(4), 506–519 (2003)
4. Blum, M., Hopper, N.J.: A secure human-computer authentication scheme. Technical report, CMU-CS-00-139, School of Computer Science, CMU (2000)
5. Bringer, J., Chabanne, H.: Trusted-HB: a low-cost version of HB^+ secure against man-in-the-middle attacks. IEEE Trans. Inf. Theory **54**(9), 4339–4342 (2008)
6. Bringer, J., Chabanne, H., Emmanuelle, D.: HB^{++}: a lightweight authentication protocol secure against some attacks. In: IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing - SecPerU 2006, pp. 28–33. IEEE (2006)
7. Duc, D.N., Kim, K.: Securing HB^+ against GRS man-in-the-middle attack. In: Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security (2007)
8. Frumkin, D., Shamir, A.: Un-trusted-HB: security vulnerabilities of trusted-HB. IACR Cryptology ePrint Archive, p. 44 (2009)
9. Gilbert, H., Robshaw, M., Sibert, H.: Active attack against HB^+: a provably secure lightweight authentication protocol. Electron. Lett. **41**(21), 1169–1170 (2005)
10. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: $HB^{\#}$: increasing the security and efficiency of HB^+. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 361–378. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_21
11. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: Good variants of HB^+ are hard to find. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 156–170. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85230-8_12
12. Hopper, N.J., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_4
13. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_18
14. Ka, A.K.: hHB: a harder HB^+ protocol. In: SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, pp. 163–169 (2015)
15. Katz, J., Shin, J.S.: Parallel and concurrent security of the HB and HB^+ protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_6
16. Leng, X., Mayes, K., Markantonakis, K.: HB-MP^+ protocol: an improvement on the HB-MP protocol. In: IEEE International Conference on RFID 2008, pp. 118–124. IEEE (2008)
17. Levieil, É., Fouque, P.-A.: An improved LPN algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006). https://doi.org/10.1007/11832072_24
18. Munilla, J., Peinado, A.: HB-MP: a further step in the HB-family of lightweight authentication protocols. Comput. Netw. **51**(9), 2262–2267 (2007)
19. Ouafi, K., Overbeck, R., Vaudenay, S.: On the security of $HB^{\#}$ against a man-in-the-middle attack. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 108–124. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_8

20. Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, J.M.E., Ribagorda, A.: Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In: Chung, K.-I., Sohn, K., Yung, M. (eds.) WISA 2008. LNCS, vol. 5379, pp. 56–68. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00306-6_5
21. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2005, pp. 84–93. ACM (2005)
22. Rizomiliotis, P., Gritzalis, S.: *GHB* $^{\#}$: a provably secure *HB*-like lightweight authentication protocol. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 489–506. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31284-7_29