Un-Trusted-HB: Security Vulnerabilities of Trusted-HB

Dmitry Frumkin and Adi Shamir

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
dmitry.frumkin@gmail.com, adi.shamir@weizmann.ac.il

January 27, 2009

Abstract

With increased use of passive RFID tags, the need for secure lightweight identification protocols arose. HB+ is one such protocol, which was proven secure in the detection-based model, but shown breakable by man-in-the-middle attacks. Trusted-HB is a variant of HB+, specifically designed to resist man-in-the-middle attacks. In this paper, we discuss several weaknesses of Trusted-HB, show that the formal security proof provided by its designers is incorrect, and demonstrate how to break it in realistic scenarios.

1 Introduction

With increased use of passive RFID tags, the need for secure lightweight identification protocols arose. One family of such protocols is based on the NP-hard problem of learning parity with noise [1, 2]. The first protocol of this kind, called HB [3], was designed for use by humans and shown to be secure only against passive adversaries. Since RFID tags, like humans, are limited in computing power, Juels and Weiss used HB as a basis to create an identification protocol for RFID, called HB+ [4], which is provably secure against active attacks in the detection-based model (defined in [4]), in which an adversary can eavesdrop on the tag-reader communication channel and communicate only with tags, but not readers, before attempting to pass an identification session posing as a tag. Katz et al. [5, 6, 7] simplified and extended the original proof of security. Nevertheless, HB+ was shown to be insecure against a stronger adversary that can perform man-in-the-middle attacks [8]. Since then, many attempts have been made to design an LPN-based protocol, secure against MIM attacks [9, 10, 11, 12, 13, 14, 15, 16], and some of them have been broken [17, 18]. This paper discusses the security of one such improvement attempt, called Trusted-HB [15]. Section 2 introduces HB+ and describes some known attacks on it; Section 3 describes the Trusted-HB proposal; Section 4 describes problems with the design of Trusted-HB and demonstrates how to break the scheme in realistic scenarios; finally, Section 5 summarizes the paper and describes possible directions for future research.

2 HB+

HB+ is a lightweight secret-key protocol proposed for RFID tag identification by Juels and Weis [4]. The publicly known parameters are: positive integers k_1, k_2, r and real values $\eta, u \in (0, 0.5)$. The reader and the tag share two secrets $\boldsymbol{x} \in_R \{0,1\}^{k_1}$ and $\boldsymbol{y} \in_R \{0,1\}^{k_2}$. The protocol proceeds in r 3-move rounds as shown in Figure 1: the tag generates a random blinding factor $\boldsymbol{b}^{(i)} \in_R \{0,1\}^{k_2}$ and sends it to the server; the server responds with a random challenge $\boldsymbol{a}^{(i)} \in_R \{0,1\}^{k_1}$; the tag computes $z^{(i)} = \boldsymbol{a}^{(i)}\boldsymbol{x}+\boldsymbol{b}^{(i)}\boldsymbol{y}+\nu^{(i)}$, where $\nu^{(i)} \sim \mathrm{Ber}_\eta$, and sends it to the server. The reader accepts the tag if the number of i's, for which $z^{(i)} \neq \boldsymbol{a}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}\boldsymbol{y}$ does not exceed ur.

Figure 1: The i'th Round of HB+

Tag (secret $\boldsymbol{x} \in_R \{0,1\}^{k_1}, \boldsymbol{y} \in_R \{0,1\}^{k_2}$)		Reader (secret $\boldsymbol{x} \in_R \{0,1\}^{k_1}, \boldsymbol{y} \in_R \{0,1\}^{k_2}$)
$\boldsymbol{b}^{(i)} \in_R \{0,1\}^{k_2}$	$\xrightarrow{\boldsymbol{b}^{(i)}}$	
	$\xleftarrow{\boldsymbol{a}^{(i)}}$	$\boldsymbol{a}^{(i)} \in_R \{0,1\}^{k_1}$
$\nu^{(i)} \sim \mathrm{Ber}_\eta$ $z^{(i)} = \boldsymbol{a}^{(i)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(i)} \cdot \boldsymbol{y} + \nu^{(i)}$	$\xrightarrow{z^{(i)}}$	$\boldsymbol{a}^{(i)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(i)} \cdot \boldsymbol{y} \overset{?}{=} z^{(i)}$

Katz et al. [5, 6, 7] showed that HB+ is asymptotically secure for any choice of $\eta \in (0, 0.5)$ and extended the proof to parallel/concurrent versions of HB+. It is possible to make the completeness and soundness errors sufficiently small by the appropriate choice of r and u. k_1 needs to be at least 80, so it is hard to guess $\boldsymbol{x}$, while k_2 is chosen to make the LPN problem with the parameters η, k_2 sufficiently hard. Several algorithms were proposed to solve the LPN problem [19, 20, 21, 22, 23]. See [22] for the state-of-the-art heuristic algorithm LF2 and the recommended parameter values for HB+.

In the prevention-based model, which allows the adversary to communicate with tags and readers at the same time, HB+ can be broken in linear time with a simple man-in-the-middle attack [8], which is referred to as the GRS attack by the first letters of the names of its authors. The adversary chooses a $\boldsymbol{\delta} \in \{0,1\}^{k_1}$ and replaces every $\boldsymbol{a}_i$ by $\boldsymbol{a}_i + \boldsymbol{\delta}$. If the identification session succeeds, he concludes that $(\boldsymbol{a}_i + \boldsymbol{\delta}) \cdot \boldsymbol{x} = \boldsymbol{a}_i \cdot \boldsymbol{x}$ and, therefore, $\boldsymbol{\delta} \cdot \boldsymbol{x} = 0$; otherwise, $\boldsymbol{\delta} \cdot \boldsymbol{x} = 1$ with overwhelming probability. Having access to k_1 sessions, the adversary repeats the above procedure with k_1 linearly independent values of $\boldsymbol{\delta}$ (e.g. the standard basis) and is able to learn the secret $\boldsymbol{x}$. Since the tag can generate any blinding factors it wants, the knowledge of $\boldsymbol{x}$ is sufficient to successfully forge an RFID tag.

3 Trusted-HB

3.1 The Design Principles of Trusted-HB

If the attacker modifies the communication, then the reader and the tag have different views of the transcript. Therefore, one way to achieve security against man-in-the-middle attacks is for the tag to send a signature of its view of the transcript. The adversary will need to replace it by the signature of the transcript as seen by the reader, which is computationally infeasible if a secure (existentially unforgeable) signature scheme is used [24]. Note that the security parameters of the signature scheme cannot be significantly relaxed. For example, consider the linear-time GRS attack on HB+ and assume that the adversary knows how to forge a signature for the modified transcript with success probability ε. Then he will repeat his attack with a fixed $\boldsymbol{\delta}$ in c/ε sessions, where c is some constant. If $\boldsymbol{\delta} \cdot \boldsymbol{x} = 0$, the adversary expects the tag to be accepted with probability about $1 - (1-\varepsilon)^{c/\varepsilon} > 1 - e^{-c}$ and otherwise rejected with overwhelming probability.

The extended identification protocol has two stages:

(i) the original identification protocol,

(ii) signing the transcript and verifying the signature.

Note that the verifier needs to check the signature only if the first stage is passed. Since the original protocol is secure in the detection-based model and transcript signing makes man-in-the-middle attacks impossible, the new protocol is secure in the prevention-based model.

Now, the challenge for RFIDs is that the signature scheme has to be implemented on the tag and, therefore, be very lightweight. Note that using a full-fledged MAC, such as SQUASH [25], in order to sign the transcript (in addition to the original identification protocol) is too resource-demanding for a passive RFID tag. In addition, such a solution would defeat the purpose, as the MAC itself can be used for identification in a challenge-response protocol - and more efficiently, since the whole transcript is longer than a single challenge. Therefore, to make this approach practical, one has to make compromises which weaken the security of the protocol.

3.2 The LFSR-based Signature Scheme

Let H be a publicly known universal family of hash functions (as defined by Carter and Wegman in [26]) from $\{0,1\}^m$ to $\{0,1\}^n$. Let $h \in_R H$ and a number of one-time pads $\boldsymbol{e}^{(i)} \in_R \{0,1\}^n$ be the secret key shared by two communicating parties. A Carter-Wegman MAC defines the signature of a message $\boldsymbol{M} \in \{0,1\}^m$ by $\boldsymbol{t} = h(\boldsymbol{M}) + \boldsymbol{e}^{(i)} \in \{0,1\}^n$, where $\boldsymbol{e}^{(i)}$ is the first unused one-time pad. Since the universal family of hash functions H is by definition perfectly balanced and thanks to the use of one-time pads, Carter-Wegman MACs are perfectly secure in the information-theoretic sense.

One way to construct a family of universal hash functions, introduced in [27], is based on $n \times m$ boolean Toeplitz matrices. These are matrices which contain a fixed value in each left-to-right diagonal, i.e. U is a Toeplitz matrix if $U_{i,j} = U_{i+k,j+k}$ for every $0 \leq i, i+k < n$ and $0 \leq j, j+k < m$. The family H can be described as a collection of such matrices, namely, every $h \in H$ corresponds to a Toeplitz matrix U, and $h(M)$ is computed as shown in Equation 1.

$$h(\boldsymbol{M}) = \underbrace{\begin{bmatrix} u_0 & u_1 & u_2 & \cdots & \ddots & \ddots & u_{m-1} \\ u_{-1} & u_0 & u_1 & \cdots & \ddots & \ddots & u_{m-2} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ u_{1-n} & u_{2-n} & u_{3-n} & \cdots & \cdots & \cdots & u_{m-n} \end{bmatrix}}_{U} \cdot \underbrace{\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ \vdots \\ M_{m-1} \end{bmatrix}}_{\boldsymbol{M}} \tag{1}$$

Hugo Krawczyk [28] suggested restricting the family of all the Toeplitz matrices to only those, whose consecutive columns can be compactly represented as the consecutive states of an LFSR with an irreducible connection polynomial p of degree n. Let f be the feedback function corresponding to p. Then $u_1 = f(u_{1-n}, \ldots, u_0), u_2 = f(u_{2-n}, \ldots, u_0, u_1)$, etc. Each function in this family can be described as $(\boldsymbol{s}, p)$, where $\boldsymbol{s} = \overline{u_0 u_{-1} \ldots u_{1-n}}$, which results in significant savings for a large m.

The LFSR-based hash family H is not perfectly balanced, so the perfect security guarantee of the original Toeplitz construction is lost; however, Krawczyk showed that it is *ε-otp-balanced* for $\varepsilon \leq \frac{m}{2^{n-1}}$, i.e.

$$\forall \boldsymbol{M} \in \{0,1\}^m, \boldsymbol{M} \neq \mathbf{0}, \boldsymbol{c} \in \{0,1\}^n, \quad \Pr_{h \in H}[h(\boldsymbol{M}) = \boldsymbol{c}] \leq \varepsilon \tag{2}$$

In addition, H is *$\oplus$-linear*, i.e.

$$\forall h \in H, \boldsymbol{M} \neq \boldsymbol{M}' \in \{0,1\}^m, \quad h(\boldsymbol{M} + \boldsymbol{M}') = h(\boldsymbol{M}) + h(\boldsymbol{M}') \tag{3}$$

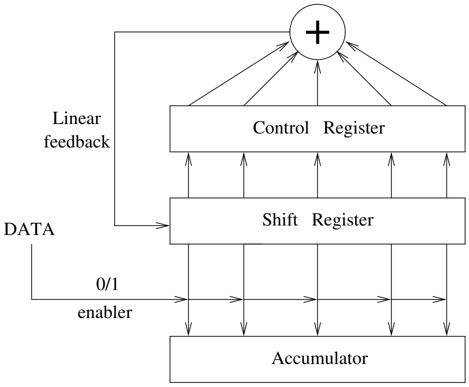

Figure 2: The LFSR-based Toeplitz hashing

Krawczyk showed that these two properties of the LFSR-based Toeplitz construction are enough to guarantee that it is *ε-otp-secure* for the same $\varepsilon \le \frac{m}{2^{n-1}}$, according to the following definition:

Definition 3.1 *Let H be a family of hash functions from $\{0,1\}^m$ to $\{0,1\}^n$. Let $h \in_R H$ and a collection of one-time pads $\boldsymbol{e}^{(i)} \in_R \{0,1\}^n$ be the secret key shared by two communicating parties. Consider the signature scheme in which the signature of an $\boldsymbol{M} \in \{0,1\}^m$ is $\boldsymbol{t} = h(\boldsymbol{M}) + \boldsymbol{e}^{(i)} \in \{0,1\}^n$, where $\boldsymbol{e}^{(i)}$ is the first unused one-time pad. Consider an all-powerful adversary that knows the H, but not the h or the $\boldsymbol{e}^{(i)}$'s and can observe any number s of message-signature pairs $(\boldsymbol{M}^{(i)}, h(\boldsymbol{M}^{(i)}) + \boldsymbol{e}^{(i)})$ for $i = 0, \ldots, s-1$. He is then given a pair $(\boldsymbol{M}^{(s)}, h(\boldsymbol{M}^{(s)}) + \boldsymbol{e}^{(s)})$ and is asked to generate a pair $(\boldsymbol{M}', h(\boldsymbol{M}') + \boldsymbol{e}^{(s)})$, where the $\boldsymbol{M}'$ was not previously signed. If the success probability over the choice of h for any such adversary is upper-bounded by ε, the hash family H is called* ε-otp-secure.

Thus, for the *ε-otp-security* of the MAC, one can set

$$n = \lceil \log(m) - \log(\varepsilon) \rceil + 1, \tag{4}$$

which is much smaller than m for large values of m and a fixed ε. The secret key of the signature scheme consists of the randomly chosen irreducible connection polynomial p, the seed $\boldsymbol{s} \in_R \{0,1\}^n$ and a number of one-time pads $\boldsymbol{e}^{(i)} \in_R \{0,1\}^n$, one per message to be authenticated. The feedback connections of the LFSR can be efficiently implemented in hardware as shown in Figure 2. The hash function starts with the secret seed $\boldsymbol{s}$ in the LFSR and $\boldsymbol{0}$ in the accumulator. For each bit of $\boldsymbol{M}$, if it is equal to 1, the value in the shift register is XOR'ed into the value of the accumulator, and then the LFSR is clocked.

3.3 The Implementation of Trusted-HB

Bringer and Chabanne [15] used the techniques described above in the design of the two-stage identification protocol Trusted-HB:

(i) the standard HB+ protocol is executed;

(ii) the Krawczyk's scheme is used by the tag to sign and by the reader to verify the integrity of the transcript of the first stage.

Let us discuss the implementation details of the signature generation:

- The reader and the tag share the connection polynomial p and the seed value $\boldsymbol{s}$ as part of the secret key in addition to $\boldsymbol{x}$ and $\boldsymbol{y}$.
- The transcript is represented as $\boldsymbol{M} = z^{(r-1)}\|\boldsymbol{a}^{(r-1)}\|\boldsymbol{b}^{(r-1)}\|\cdots\|z^{(0)}\|\boldsymbol{a}^{(0)}\|\boldsymbol{b}^{(0)}$; hence, $m = |\boldsymbol{M}| = r(k_1 + k_2 + 1)$.
- Since the tag cannot store the whole transcript $\boldsymbol{M}$, the signature $\boldsymbol{t}$ is constructed progressively along the way, with the LFSR being clocked $k_1 + k_2 + 1$ times per round of HB+. The tag uses separate circuitry to generate the signature in parallel with the HB+ computations.
- Since a tag can participate in many identification sessions, there is no way it could store enough one-time pads for all of them. Normally, the noise $\boldsymbol{v}$ is generated in the tag from a physical source of randomness, which cannot be replicated by the reader. This source of randomness can be used to generate one-time pads; however, we need to make sure that the reader can also compute them, while keeping the scheme safe from desynchronization attacks. Bringer and Chabanne suggest the novel technique of generating the one-time pad $\boldsymbol{e}$ from the LPN noise $\boldsymbol{v}$ (which becomes known to the reader by locating the errors in the values sent by the tag). Since $\boldsymbol{v}$ is unbalanced, both parties have to use a randomness extractor E to balance the output: $\boldsymbol{e} = E(\boldsymbol{v})$. Since the reader knows $\boldsymbol{x}$ and $\boldsymbol{y}$, it can compute $\boldsymbol{v} = A\boldsymbol{x} + B \cdot \boldsymbol{y} + \boldsymbol{z}$ and, hence, $\boldsymbol{e}$; however, an attacker cannot identify the locations of the errors, since this is equivalent to breaking the security of the scheme by solving a system of error-free linear equations.
- To avoid having to store all of $\boldsymbol{v}$ in the tag, the randomness extractor E should be such that the tag can compute $\boldsymbol{e}$ (and hence $\boldsymbol{t}$) progressively. Bringer and Chabanne suggest using the von Neumann procedure [29]: for each bit pair $\left(v^{(2i)}, v^{(2i+1)}\right)$, if $v^{(2i)} \neq v^{(2i+1)}$, then the next bit of $\boldsymbol{e}$ is $v^{(2i)}$; otherwise, the pair is not used. Let $F : \{0,1\}^* \to \{0,1\}^*$ denote the von Neumann procedure; it can be expressed recursively as:

  $$F(\boldsymbol{v}) = \begin{cases} F(\overline{\ldots v_4 v_3 v_2}), & \text{if } v_0 = v_1 \\ F(\overline{\ldots v_4 v_3 v_2})\|v_0, & \text{if } v_0 \neq v_1 \end{cases} \tag{5}$$

  While the length of $F(\boldsymbol{v})$ is variable, only the first n bits of the output are computed and used.
- Bringer and Chabanne propose the same parameter values as suggested for HB+: $\eta = 0.25, k_1 = 80, k_2 = 512, r = 1164, u = 0.348$, so $2^{19} < m = r(k_1 + k_2 + 1) < 2^{20}$; therefore, according to (4), to achieve *ε-otp-security* for $\varepsilon = 2^{-80}$, they fix $n = \lceil \log(m) - \log(\varepsilon) \rceil + 1 = 101$. For $\eta = 0.25$, the mean output length of the von Neumann procedure is 218, where the probability of having fewer than $n = 101$ bits is less than 2^{-72} (if this ever happens, the identification restarts).

4 Attacks on Trusted-HB

4.1 Weaknesses of Trusted-HB

As we discussed in Section 3.1, using a signature scheme to authenticate the transcript should secure the identification scheme against man-in-the-middle adversaries. We claim, however, that Trusted-HB cuts corners in several places, and some of these modifications make the protocol insecure:

- h cannot be kept completely secret in practice, while the security of Trusted-HB relies on h being part of the secret key and completely unknown to the adversary. The main reason the LFSR-based construction was proposed is that it is easy to implement in hardware, which means that the connections corresponding to the feedback polynomial p are likely to be hardwired (note that it is important that the LFSR shifting be fast, as the complexity of a single shift is multiplied by m). It is impractical to have different p's hardwired in different tags, and thus millions of tags are likely to have the same feedback. Finally, we cannot rely on security by obscurity (which is a bad idea anyway) as it has been demonstrated that the circuitry of an RFID tag can be deduced by using just an optical microscope [30]. Therefore, while the unique seed $\boldsymbol{s}$ can be kept secret, we have to assume that in practice the p will become publicly known.

- $\boldsymbol{e}$ is not a real one-time pad. As mentioned earlier, the *ε-otp-security* of the signature scheme depends on $\boldsymbol{e}$ being a one-time pad; however, it is impossible to store one-time pads in the tag. Recycling $\boldsymbol{\nu}$ to generate $\boldsymbol{e}$, which looked like a clever implementation trick, makes the protocol insecure.

- For small values of η, the chances that E returns sufficiently many random bits for $\boldsymbol{e}$ are not so high. The rate of the extraction procedure, defined as $R(\eta) = \limsup_{r\to\infty} \frac{1}{r}\mathrm{E}[|E(\nu_1, \cdots, \nu_r)|]$, at best can approach the entropy bound: $R(\eta) \le h(\eta) := -\eta \log_2 \eta - (1-\eta)\log_2(1-\eta)$, which decreases with η (see [31] for an improvement of the von Neumann procedure that approaches the bound). For example, for $\eta = 0.05, k_1 = 80, k_2 = 768, r = 249$ (chosen according to the recommendations made in [22]), we have $m = 211,401$ and $r \cdot R(\eta) \approx 71$. n must be way below that (unless we want to restart the identification every other time) and so $\varepsilon = \frac{m}{2^{n-1}} > 2^{-62}$. Also, whenever an identification session is restarted after (or during) the HB+ stage of Trusted-HB (because the tag cannot produce an $\boldsymbol{e}$ of length n), the information that E cannot extract n random bits from the noise $\boldsymbol{\nu}$, may be useful to the adversary.

The last weakness above may be not as serious as the other two, as it might be dealt with by disallowing small values of η, increasing the round complexity r, or by specifying explicitly how and when the identification session is restarted, so that the adversary cannot benefit much from knowing that the noise $\boldsymbol{\nu}$ does not satisfy certain properties. Nevertheless, this flaw was not addressed in the original proposal and is worth mentioning.

We now show how to use the first two weaknesses above to attack Trusted-HB. In the realistic attack scenario in which the connection polynomial p of the LFSR is known, there are efficient man-in-the-middle attacks described in Section 4.2. The fact that $\boldsymbol{e}$ is not a real one-time pad gives rise to a slower but completely passive attack, which is described in Section 4.3.

4.2 MIM Attacks

The LFSR used for signing the transcript starts with $\boldsymbol{s} = \overline{u_0 u_{-1} \ldots u_{1-n}}$ as the seed, so for every $i \ge 1-n$, there is a linear dependence $u_{i+j_0} + u_{i+j_1} + \cdots + u_{i+j_{l-1}} + u_{i+j_l} = 0$, where l is the number of taps in the LFSR, $j_0, \ldots, j_{l-1}$ are the tap positions, and $0 = j_0 < j_1 < \cdots < j_l = n$. Let $\boldsymbol{u}^{(i)}$ denote the i'th column of the Toeplitz matrix U. Since $\boldsymbol{u}^{(i)}$'s are the consecutive states of the LFSR, the same recurrence applies to them:

$$\forall i \ge 1-n, \qquad \boldsymbol{u}^{(i+j_0)} + \boldsymbol{u}^{(i+j_1)} + \boldsymbol{u}^{(i+j_2)} + \cdots + \boldsymbol{u}^{(i+j_{l-1})} + \boldsymbol{u}^{(i+n)} = \boldsymbol{0} \tag{6}$$

Let us define $\boldsymbol{\Delta} \in \{0,1\}^{n+1}$ by

$$\Delta_i = \begin{cases} 1, & \text{if } i \in \{j_0, \ldots, j_l\} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

and $\text{Shift}^q : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^q$ by

$$\text{Shift}^q(\boldsymbol{w}, d) = \underbrace{0\ldots0\|\boldsymbol{w}\|\underbrace{0\ldots0}_{d}}_{q} \tag{8}$$

where $q \geq d + |\boldsymbol{w}|$.

Definition 4.1 *Let H be the LFSR-generated hash family. We call $\boldsymbol{\psi} \in \{0,1\}^m$ a complete MIM pattern for p if $h(\boldsymbol{\psi}) = \mathbf{0}$ for every $h \in H$ that uses p as the connection polynomial. We call $\mathbf{w}$ a (simple) MIM pattern for p if $v_0 = v_{|\boldsymbol{w}|-1} = 1$ and $Shift^m(\boldsymbol{w}, 0)$ is a complete MIM pattern for p.*

By definition, a MIM pattern $\boldsymbol{w}$ satisfies $U \cdot \text{Shift}^m(\boldsymbol{w}, 0) = \mathbf{0}$, where the Toeplitz matrix U can be generated with any seed $\boldsymbol{s} \in \{0,1\}^n$; therefore, $\boldsymbol{w}$ must satisfy $U \cdot \text{Shift}^m(\boldsymbol{w}, d) = \mathbf{0}$ for every $d \leq m - |\boldsymbol{w}|$, i.e. every such $\text{Shift}^m(\boldsymbol{w}, d)$ is a complete MIM pattern. Furthermore, by $\oplus$-*linearity* of H, any linear combination of complete MIM patterns is a complete MIM pattern, so one way to construct a complete MIM pattern from a simple MIM pattern is by taking a linear combination of $\text{Shift}^m(\boldsymbol{w}, d)$'s for various values of d.

By (6), $\boldsymbol{\Delta}$ is a MIM pattern of length $n + 1$. Since the adversary knows the p, he can also find other, longer MIM patterns offline by employing various methods used in correlation attacks (see, e.g., [32]).

Let us use the following notation:

- $\boldsymbol{b}^{(i)}, \boldsymbol{a}^{(i)}, z^{(i)}$ are the values seen by the tag at round $i \in \{0, \ldots, r-1\}$; $\boldsymbol{M}$ is the tag's version of the transcript; $\boldsymbol{v}$ is the noise generated by the tag; $\boldsymbol{e}$ is the one-time pad and $\boldsymbol{t}$ is the signature computed by the tag.

- $\hat{\boldsymbol{b}}^{(i)}, \hat{\boldsymbol{a}}^{(i)}, \hat{z}^{(i)}$ are the values seen by the reader at round $i \in \{0, \ldots, r-1\}$; $\hat{\boldsymbol{M}}$ is the reader's version of the transcript; $\hat{\boldsymbol{v}}$ is the noise, $\hat{\boldsymbol{e}}$ is the one-time pad, and $\hat{\boldsymbol{t}}$ is the signature computed by the reader.

- $\boldsymbol{M}'$ is the transcript composed of the values actually sent by the two parties (the $\boldsymbol{b}^{(i)}$'s, $\hat{\boldsymbol{a}}^{(i)}$'s, and $\boldsymbol{z}^{(i)}$'s) and seen by the man-in-the-middle adversary.

- $\bar{\boldsymbol{b}}^{(i)} = \boldsymbol{b}^{(i)} + \hat{\boldsymbol{b}}^{(i)}$, $\bar{\boldsymbol{a}}^{(i)} = \boldsymbol{a}^{(i)} + \hat{\boldsymbol{a}}^{(i)}$, $\bar{z}^{(i)} = z^{(i)} + \hat{z}^{(i)}$ for every $i \in \{0, \ldots, r-1\}$. $\bar{\boldsymbol{M}}$ is the transcript of all the changes made by the adversary, i.e. the $\bar{\boldsymbol{b}}^{(i)}$'s, $\bar{\boldsymbol{a}}^{(i)}$'s, and $\bar{z}^{(i)}$'s. Thus, $\bar{\boldsymbol{M}} = \boldsymbol{M} + \hat{\boldsymbol{M}}$, and the transcript of all the values received by the tag and the reader is $\boldsymbol{M}' + \bar{\boldsymbol{M}}$.

The key observation is that if $\bar{\boldsymbol{M}}$ is a complete MIM pattern, then by $\oplus$-*linearity* of H, $h(\boldsymbol{M}) = h(\hat{\boldsymbol{M}})$. This means that if the adversary uses such an $\bar{\boldsymbol{M}}$ to modify the messages sent during the first stage of Trusted-HB, the identification session results in 'accept' if the HB+ stage is passed and $\boldsymbol{e} = \hat{\boldsymbol{e}}$.

The first attack below is similar to the GRS attack on HB+ considered in Section 2; in particular, acceptance of the modified transcript in the HB+ stage is (with overwhelming probability) synonymous with $\boldsymbol{v} = \hat{\boldsymbol{v}}$. Since $\boldsymbol{v} = \hat{\boldsymbol{v}} \Rightarrow \boldsymbol{e} = \hat{\boldsymbol{e}}$, acceptance in the full Trusted-HB protocol is (with overwhelming probability) synonymous with $\boldsymbol{v} = \hat{\boldsymbol{v}}$ and the attack can be carried out in variants of Trusted-HB that utilize any extraction procedure for E. The second attack is such that the HB+ stage almost always succeeds. Based on E being the von Neumann's procedure, with overwhelming probability $\boldsymbol{e} = \hat{\boldsymbol{e}} \Rightarrow \boldsymbol{v} = \hat{\boldsymbol{v}}$; therefore, acceptance is (with overwhelming probability) synonymous with $\boldsymbol{v} = \hat{\boldsymbol{v}}$.

The two attacks allow the adversary to retrieve $\boldsymbol{x}$ and $\boldsymbol{y}$. We will next show in 4.2.3 how to learn the last remaining secret $\boldsymbol{s}$ needed to counterfeit a tag. Finally, we will describe in 4.2.4 a toy example that demonstrates these two attacks in practice.

4.2.1 The First Attack

When $n < k_1 + k_2 + 1$, there are MIM patterns, such as Δ, that have length $\leq k_1 + k_2 + 1$. Let us fix any such pattern $\boldsymbol{w}$, any $j \in \{0, \ldots, k_1 + k_2 + 1 - |\boldsymbol{w}|\}$, and let $\mathring{\boldsymbol{w}} = \mathrm{Shift}^{k_1+k_2+1}(\boldsymbol{w}, j)$. Let $\bar{\boldsymbol{b}} = \mathring{w}_{k_2-1} \ldots \mathring{w}_0, \bar{\boldsymbol{a}} = \mathring{w}_{k_1+k_2-1} \ldots \mathring{w}_{k_2}$, and $\bar{z} = \mathring{w}_{k_1+k_2}$, so $\bar{z} \| \bar{\boldsymbol{a}} \| \bar{\boldsymbol{b}} = \mathring{\boldsymbol{w}}$.

Definition 4.2 *We say that the adversary "applies" the change $\mathring{\boldsymbol{w}}$ to round i of the HB+ stage of the protocol, if he replaces $\boldsymbol{b}^{(i)}$ by $\boldsymbol{b}^{(i)} + \bar{\boldsymbol{b}}$, $\hat{\boldsymbol{a}}^{(i)}$ by $\hat{\boldsymbol{a}}^{(i)} + \bar{\boldsymbol{a}}$, and $z^{(i)}$ by $z^{(i)} + \bar{z}$.*

The adversary "applies" $\mathring{\boldsymbol{w}}$ to every round of the protocol, resulting in the following collection of transcripts:

$$\begin{array}{rlllllllllll}
\boldsymbol{M} = & z^{(r-1)} & \| & \boldsymbol{a}^{(r-1)} & \| & \boldsymbol{b}^{(r-1)} & \|\cdots\| & z^{(0)} & \| & \boldsymbol{a}^{(0)} & \| & \boldsymbol{b}^{(0)} \\
\hat{\boldsymbol{M}} = & \hat{z}^{(r-1)} & \| & \hat{\boldsymbol{a}}^{(r-1)} & \| & \hat{\boldsymbol{b}}^{(r-1)} & \|\cdots\| & \hat{z}^{(0)} & \| & \hat{\boldsymbol{a}}^{(0)} & \| & \hat{\boldsymbol{b}}^{(0)} \\
\bar{\boldsymbol{M}} = & z^{(r-1)} + & \hat{z}^{(r-1)} \| \boldsymbol{a}^{(r-1)} + & & \hat{\boldsymbol{a}}^{(r-1)} \| \boldsymbol{b}^{(r-1)} + & \hat{\boldsymbol{b}}^{(r-1)} & \|\cdots\| z^{(0)} + & \hat{z}^{(0)} \| \boldsymbol{a}^{(0)} + & & \hat{\boldsymbol{a}}^{(0)} \| \boldsymbol{b}^{(0)} + & \hat{\boldsymbol{b}}^{(0)} & \\
= & \bar{z} & \| & \bar{\boldsymbol{a}} & \| & \bar{\boldsymbol{b}} & \|\cdots\| & \bar{z} & \| & \bar{\boldsymbol{a}} & \| & \bar{\boldsymbol{b}} \\
= & & & \mathring{\boldsymbol{w}} & & & \|\cdots\| & & & \mathring{\boldsymbol{w}} & &
\end{array}$$

Note that $\bar{\boldsymbol{M}}$ is a complete MIM pattern since it is a linear combination of complete MIM patterns:

$$\bar{\boldsymbol{M}} = \sum_{i=0}^{r-1} \mathrm{Shift}^{m}(\boldsymbol{w}, j + i(k_1 + k_2 + 1))$$

Also,

$$\begin{aligned}
\boldsymbol{v} = \hat{\boldsymbol{v}} &\Leftrightarrow \forall i = 0, \ldots, r-1, \quad \boldsymbol{b}^{(i)}\boldsymbol{y} + \hat{\boldsymbol{a}}^{(i)}\boldsymbol{x} + z^{(i)} = \hat{\boldsymbol{b}}^{(i)}\boldsymbol{y} + \boldsymbol{a}^{(i)}\boldsymbol{x} + \hat{z}^{(i)} \\
&\Leftrightarrow \forall i = 0, \ldots, r-1, \quad (\boldsymbol{b}^{(i)} + \hat{\boldsymbol{b}}^{(i)})\boldsymbol{y} + (\boldsymbol{a}^{(i)} + \hat{\boldsymbol{a}}^{(i)})\boldsymbol{x} + (z^{(i)} + \hat{z}^{(i)}) = 0 \\
&\Leftrightarrow \bar{\boldsymbol{b}}\boldsymbol{y} + \bar{\boldsymbol{a}}\boldsymbol{x} + \bar{z} = 0
\end{aligned}$$

Thus, if $\bar{\boldsymbol{b}}\boldsymbol{y} + \bar{\boldsymbol{a}}\boldsymbol{x} + \bar{z} = 0$, then $\boldsymbol{v} = \hat{\boldsymbol{v}}$ and both stages of Trusted-HB are passed (with overwhelming probability), resulting in 'accept'. If $\bar{\boldsymbol{b}}\boldsymbol{y} + \bar{\boldsymbol{a}}\boldsymbol{x} + \bar{z} = 1$, then the HB+ stage fails with overwhelming probability, resulting in 'reject'. Since this can be done for $j = 0, \ldots, k_1 + k_2 + 1 - |\boldsymbol{w}|$, we can get $k_1 + k_2 + 2 - |\boldsymbol{w}|$ linear equations in the bits of $\boldsymbol{x}$ and $\boldsymbol{y}$. Furthermore, since the patterns $\mathring{\boldsymbol{w}}$ are all linearly independent as consecutive linear shifts of the same $\boldsymbol{w}$, all of these equations are useful.

To demonstrate this attack, consider the parameter values $k_1 = 80, k_2 = 512, n = 101$ proposed by Bringer and Chabanne. Take the MIM pattern $\boldsymbol{\Delta}$ as defined in (7). Since $|\boldsymbol{\Delta}| = n + 1 = 102$, the adversary can get $80 + 512 + 2 - 102 = 492$ linear equations. He needs only $|\boldsymbol{\Delta}| - 2 = n - 1 = 100$ more linear equations to efficiently solve for all the bits of $\boldsymbol{x}$ and $\boldsymbol{y}$ by Gaussian elimination. These equations can be obtained by utilizing a different short MIM pattern or by using these linear relationships to simplify the noisy parity equations and efficiently solving LPN for the greatly improved parameters $\eta = 0.25, k = 100$ using the methods of [23] or [22].

If the length of the MIM pattern is $\leq k_2$, a small variation of the above attack is for the adversary to target only $\boldsymbol{y}$. For the proposed parameter values, $|\boldsymbol{\Delta}| = n + 1 = 102 \leq k_2 = 512$, so the adversary may consider shifts with $j = 0, 1, \ldots, k_2 - |\boldsymbol{\Delta}| = 410$ to get 411 linearly independent linear equations in the 512 bits of $\boldsymbol{y}$. By using other short MIM patterns, the adversary can learn $\boldsymbol{y}$ with fewer man-in-the-middle interactions than in the original attack. Once he knows $\boldsymbol{y}$, the adversary can compute $\boldsymbol{b}^{(i)}\boldsymbol{y}$ for any $\boldsymbol{b}^{(i)}$ and thus eliminate its masking effect. With a simple (not MIM) active attack, the adversary can find $\boldsymbol{\delta x}$ for any $\boldsymbol{\delta} \in \{0, 1\}^{k_1}$ by sending the same challenge $\boldsymbol{\delta}$ sufficiently many times and eliminating noise by majority voting. Thus, he can learn every bit of $\boldsymbol{x}$ by having the $\boldsymbol{\delta}$ go over the standard base of $\{0, 1\}^{k_1}$.

A multi-round version of this attack technique can be used with MIM patterns that are longer than $k_1 + k_2 + 1$ bits as long as their length $\leq c(k_1 + k_2 + 1)$, where $c|r$. For example, suppose that $\boldsymbol{w}$ is a MIM pattern of length $|\boldsymbol{w}| \leq c(k_1 + k_2 + 1)$. Let $\mathring{\boldsymbol{w}} = \text{Shift}^{c(k_1+k_2+1)}(\boldsymbol{w}, j)$ for a fixed $j \in \{0, 1, \ldots, c(k_1 + k_2 + 1) - |\boldsymbol{w}|\}$, and for $i = 0, 1, \ldots, c-1$, let $\mathring{\boldsymbol{w}}^{(i)} = \overline{\mathring{w}_{(i+1)(k_1+k_2+1)-1} \ldots \mathring{w}_{i(k_1+k_2+1)}}$, so $\mathring{\boldsymbol{w}} = \mathring{\boldsymbol{w}}^{(c-1)}|| \cdots ||\mathring{\boldsymbol{w}}^{(0)}$. For $i = 0, \ldots, c-1$, let $\bar{\boldsymbol{b}}^{(i)} = \overline{\mathring{w}^{(i)}_{k_2-1} \ldots \mathring{w}^{(i)}_0}$, $\bar{\boldsymbol{a}}^{(i)} = \overline{\mathring{w}^{(i)}_{k_1+k_2-1} \ldots \mathring{w}^{(i)}_{k_2}}$, $\bar{z}^{(i)} = \mathring{w}^{(i)}_{k_1+k_2}$, so $\bar{z}^{(i)}||\bar{\boldsymbol{a}}^{(i)}||\bar{\boldsymbol{b}}^{(i)} = \mathring{\boldsymbol{w}}^{(i)}$.

For every round $i = 0, \ldots, r - 1$ of the HB+ phase, the adversary "applies" $\mathring{\boldsymbol{w}}^{(i \bmod c)}$ to that round. For example, for $c = 2$, the transcripts are modified in the following way:

$$\begin{array}{rlcccccccccccc}
\boldsymbol{M} & = \cdots || & z^{(1)} & || & \boldsymbol{a}^{(1)} & || & \boldsymbol{b}^{(1)} & || & z^{(0)} & || & \boldsymbol{a}^{(0)} & || & \boldsymbol{b}^{(0)} \\
\hat{\boldsymbol{M}} & = \cdots || & \hat{z}^{(1)} & || & \hat{\boldsymbol{a}}^{(1)} & || & \hat{\boldsymbol{b}}^{(1)} & || & \hat{z}^{(0)} & || & \hat{\boldsymbol{a}}^{(0)} & || & \hat{\boldsymbol{b}}^{(0)} \\
\bar{\boldsymbol{M}} & = \cdots || & z^{(1)} + \hat{z}^{(1)} & || & \boldsymbol{a}^{(1)} + \hat{\boldsymbol{a}}^{(1)} & || & \boldsymbol{b}^{(1)} + \hat{\boldsymbol{b}}^{(1)} & || & z^{(0)} + \hat{z}^{(0)} & || & \boldsymbol{a}^{(0)} + \hat{\boldsymbol{a}}^{(0)} & || & \boldsymbol{b}^{(0)} + \hat{\boldsymbol{b}}^{(0)} \\
 & = \cdots || & \bar{z}^{(1)} & || & \bar{\boldsymbol{a}}^{(1)} & || & \bar{\boldsymbol{b}}^{(1)} & || & \bar{z}^{(0)} & || & \bar{\boldsymbol{a}}^{(0)} & || & \bar{\boldsymbol{b}}^{(0)} \\
 & = \cdots || & & & \mathring{\boldsymbol{w}}^{(1)} & & & || & & & \mathring{\boldsymbol{w}}^{(0)} & &
\end{array}$$

Now, if $\bar{\boldsymbol{a}}^{(i)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(i)}\boldsymbol{y} + \bar{z}^{(i)} = 0$ for every $i = 0, \ldots, c-1$, then $\boldsymbol{v} = \hat{\boldsymbol{v}}$ and the tag is accepted with overwhelming probability. If for some i, $\bar{\boldsymbol{a}}^{(i)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(i)}\boldsymbol{y} + \bar{z}^{(i)} = 1$, then $|\boldsymbol{v} + \hat{\boldsymbol{v}}| = dr/c$, where d is the number of such i's. If d is small (e.g. 1) and c is large (e.g. $r/2$), the slightly increased noise may be insufficient to guarantee a failure in the HB+ part with high probability. Normally, if we only consider $c \ll r$, the probability of failure is noticeable, so the adversary can use the same $\mathring{\boldsymbol{w}}$ for a number of sessions, and if the tag is still accepted in all of them, conclude that $\bar{\boldsymbol{a}}^{(i)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(i)}\boldsymbol{y} + \bar{z}^{(i)} = 0$ for every $i = 0, \ldots, c - 1$. Of course, the knowledge of the extraction procedure E is helpful since if $\boldsymbol{v} \neq \hat{\boldsymbol{v}}$, then the second stage of Trusted-HB may result in rejection. We discuss this point in Section 4.2.2.

As before, it is possible to target only $\boldsymbol{y}$. In addition, since each $\boldsymbol{w}$ is broken into pieces that are "applied" to consecutive rounds, the adversary may target only $\boldsymbol{x}$; however, it is not clear how this variation would be useful: to learn $\boldsymbol{s}$, the adversary probably needs to learn $\boldsymbol{y}$ first anyway.

Since the probability that c random equations hold simultaneously is $1/2^c$, the expected number of linear equations per $\mathring{\boldsymbol{w}}$ is only $c/2^c$. While this variant appears somewhat less efficient than the original MIM attack with short MIM patterns, the adversary may use it if there are no or not enough short patterns (e.g. when $n + 1 > k_1 + k_2 + 1$).

4.2.2 The Second Attack

Consider an attacker who interferes with just a few rounds of the HB+ stage of the protocol, but makes sure that $h(\boldsymbol{M}) = h(\hat{\boldsymbol{M}})$. The modified transcript is still accepted with a high probability because the difference between $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ is very small. However, in Trusted-HB even a small such discrepancy may lead, with a high probability, to $\boldsymbol{e} \neq \hat{\boldsymbol{e}}$, so $\boldsymbol{t} \neq \hat{\boldsymbol{t}}$ and the tag is rejected.

Let us focus on the von Neumann procedure used in Trusted-HB. Assume that the adversary finds a MIM pattern $\boldsymbol{w}$ of length $\leq k_1 + k_2 + 1$ and "applies" the corresponding $\mathring{\boldsymbol{w}}$ only to the first round of HB+. Then weight$(\bar{\boldsymbol{v}}) \leq 1$, so the HB+ stage is still passed with very high probability. If $\bar{\boldsymbol{a}}\boldsymbol{x} + \bar{\boldsymbol{b}}\boldsymbol{y} + \bar{z} = 0$, then

$\boldsymbol{v} = \hat{\boldsymbol{v}}$, so $\boldsymbol{t} = \hat{\boldsymbol{t}}$ and the tag is accepted. If $\bar{\boldsymbol{a}}\boldsymbol{x} + \bar{\boldsymbol{b}}\boldsymbol{y} + \bar{z} = 1$, then $v_0 \neq \hat{v}_0$ and $v_i = \hat{v}_i$ for every $i > 0$. Let us refer to Equation 5 that defines the von Neumann procedure F. If $v_0 = v_1$, then $\hat{v}_0 \neq \hat{v}_1$, so $F(\hat{\boldsymbol{v}}) = F(\overline{\hat{v}_{r-1} \dots \hat{v}_3\hat{v}_2})\|\hat{v}_0$ and $F(\boldsymbol{v}) = F(\overline{v_{r-1} \dots v_3 v_2})$. Since $\overline{\hat{v}_{r-1} \dots \hat{v}_3\hat{v}_2} = \overline{v_{r-1} \dots v_3 v_2}$, we get $F(\hat{\boldsymbol{v}}) = F(\boldsymbol{v})\|\hat{v}_0$. Thus, $\hat{\boldsymbol{e}} = \overline{e_{n-2} \dots e_1 e_0 \hat{v}_0}$ is different from $\boldsymbol{e}$, unless $e_{n-1} = e_{n-2} = \cdots = e_0 = \hat{v}_0$, which is extremely unlikely. Likewise, if $v_0 \neq v_1$, then with overwhelming probability, $\boldsymbol{e} \neq \hat{\boldsymbol{e}}$. Since $h(\boldsymbol{M}) = h(\hat{\boldsymbol{M}})$ (because $\boldsymbol{w}$ is a MIM pattern), we conclude that with overwhelming probability, acceptance means $\bar{\boldsymbol{a}}\boldsymbol{x} + \bar{\boldsymbol{b}}\boldsymbol{y} + \bar{z} = 0$ and rejection means $\bar{\boldsymbol{a}}\boldsymbol{x} + \bar{\boldsymbol{b}}\boldsymbol{y} + \bar{z} = 1$. The rest of the analysis is the same as in Section 4.2.1, so the only difference is that the adversary applies $\mathring{\boldsymbol{w}}$ only to the first round instead of to all the rounds. In particular, the adversary may target only $\boldsymbol{y}$ if $|\boldsymbol{w}| < k_2$.

This attack may be more useful than the attack of Section 4.2.1 when the adversary has to use a MIM pattern $\boldsymbol{w}$ of length $\geq k_1 + k_2 + 1$ and there is no c such that $|\boldsymbol{w}| \leq c(k_1 + k_2 + 1)$ and $c|r$, for example, if $n + 1 > k_1 + k_2 + 1$ and r is prime. Let $c = \lceil m/|\boldsymbol{w}| \rceil$ and define $\mathring{\boldsymbol{w}} = \mathring{\boldsymbol{w}}^{(c-1)}\| \cdots \|\mathring{\boldsymbol{w}}^{(0)}$ and $\boldsymbol{b}^{(i)}, \boldsymbol{a}^{(i)}, z^{(i)}$ for $i = 0, \dots, c-1$ as in Section 4.2.1. Now the adversary "applies" each $\mathring{\boldsymbol{w}}^{(i)}$ only to rounds 0 to $c - 1$. For $c \ll r$, the tag will still have a high probability of passing the HB+ stage since weight$(\bar{\boldsymbol{v}}) \leq c$. Let d be the smallest even number, such that $v_i = \hat{v}_i$ for all $i \geq d$. Then $F(\boldsymbol{v}) = F(\overline{v_{r-1} \dots v_d})\|F(\overline{v_{d-1} \dots v_0})$ and $F(\hat{\boldsymbol{v}}) = F(\overline{v_{r-1} \dots v_d})\|F(\overline{\hat{v}_{d-1} \dots \hat{v}_0})$, so $F(\boldsymbol{v}) = F(\hat{\boldsymbol{v}})$ is almost always equivalent to $F(\overline{v_{d-1} \dots v_0}) = F(\overline{\hat{v}_{d-1} \dots \hat{v}_0})$, which is quite unlikely (e.g. $\Pr[|F(\overline{v_{d-1} \dots v_0})| = |F(\overline{\hat{v}_{d-1} \dots \hat{v}_0})|] \leq 1/2$). By using the same MIM pattern a constant number of times, the adversary can be confident that the c linear equations ($\bar{\boldsymbol{a}}^{(i)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(i)}\boldsymbol{y} + \bar{z}^{(i)} = 0$ for every $i = 0, \dots, c-1$) hold if and only if all the sessions result in acceptance.

4.2.3 Completion of the Attacks With Known x, y, p

Once the adversary knows $\boldsymbol{x}$ and $\boldsymbol{y}$, he can always deduce $\boldsymbol{v}$ (and hence $\boldsymbol{e}$) from the transcript and learn $\boldsymbol{s}$ via a passive attack. For each observed session, the adversary can convert $U\boldsymbol{M} = \boldsymbol{t} + \boldsymbol{e}$ (Equation 1), where the $\boldsymbol{M}, \boldsymbol{t}$ and $\boldsymbol{e}$ are known, into a system of linear equations in the bits of $\boldsymbol{s}$ in time $O(mn^2)$ by replacing each of the mn entries of U by a linear combination of the n bits of $\boldsymbol{s}$. With $O(1)$ sessions, the adversary collects n linearly independent linear equations in the bits of $\boldsymbol{s}$ and solves for $\boldsymbol{s}$ by Gaussian elimination in time $O(n^3)$.

When the recommended parameter values are used, the query complexity of the MIM attacks of Sections 4.2.1 and 4.2.2 is $O(k_1 + k_2) = O(2^9)$. Their time complexity is dominated by the $O(mn^2) = O(2^{33})$ steps to recover $\boldsymbol{s}$. The memory is needed mostly to store the systems of linear equations, so the memory complexity of the attacks is $O(\max((k_1 + k_2)^2, n^2)) = O(2^{18})$.

4.2.4 A Toy Example

Since demonstrating an attack on Trusted-HB with the recommended parameter values would take a lot of space, we will consider a toy example with unrealistically small parameter values purely for illustrative purposes. Let $k_1 = 4, k_2 = 10, r = 40, \eta = 0.25, n = 10, p(X) = X^{10} + X^3 + 1$, which corresponds to $u_{i+10} = u_{i+7} + u_i$ for $-9 = 1 - n \leq i \leq m - 1 = r(k_1 + k_1 + 1) - 1 = 599$. Let the secrets be $\boldsymbol{x} = 0101_2, \boldsymbol{y} = 1100101110_2, \boldsymbol{s} = 1110001001_2$ and consider the transcript M which is summarized in Table 1. In the 0'th round, the tag sends $\boldsymbol{b}^{(0)} = 1100010100_2$, the reader replies with $\boldsymbol{a}^{(0)} = 1000_2$, the tag generates $v_0 = 0$ and sends back $z_0 = \boldsymbol{x}\boldsymbol{a}^{(0)} + \boldsymbol{y}\boldsymbol{b}^{(0)} + v_0 = 1$, etc.

Table 1: Transcript Summary (in binary)

Round i	$\boldsymbol{b}^{(i)}$	$\boldsymbol{a}^{(i)}$	$\boldsymbol{z}^{(i)}$	$(\nu^{(i)})$
0	1100010100	1000	1	(0)
1	0101101110	0011	1	(1)
2	1001110110	1010	0	(0)
3	0001001111	0111	1	(0)
4	0010111011	0001	1	(1)
5	1101110100	0111	0	(0)
6	0100010001	1100	0	(0)
7	1101110111	0100	1	(1)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

The transcript consists of $m = 600$ bits:

$$\begin{aligned} \boldsymbol{M} = \ldots \underbrace{1}_{z^{(3)}} \underbrace{0111}_{\boldsymbol{a}^{(3)}} \underbrace{0001001111}_{b^{(3)}} \underbrace{0}_{z^{(2)}} \underbrace{1010}_{\boldsymbol{a}^{(2)}} \underbrace{1001110110}_{b^{(2)}} \\ \underbrace{1}_{z^{(1)}} \underbrace{0011}_{\boldsymbol{a}^{(1)}} \underbrace{0101101110}_{b^{(1)}} \underbrace{1}_{z^{(0)}} \underbrace{1000}_{\boldsymbol{a}^{(0)}} \underbrace{1100010100_2}_{b^{(0)}} \end{aligned} \tag{9}$$

The noise generated by the tag has $r = 40$ bits: $\boldsymbol{\nu} = \ldots 10010010_2$, so, according to the von Neumann procedure, $\boldsymbol{e} = \ldots 010_2$. The first few columns of the LFSR-generated Toeplitz matrix U, written as a sequence of (transposed) binary numbers with the tap bits underlined, are:

$$\begin{aligned} \boldsymbol{s} = \boldsymbol{s}^{(0)} &= 11\underline{1}000100\underline{1}_2 \\ \boldsymbol{s}^{(1)} &= 01\underline{1}100010\underline{0}_2 \\ \boldsymbol{s}^{(2)} &= 10\underline{1}110001\underline{0}_2 \\ &\text{etc.} \end{aligned}$$

Then

$$\begin{aligned} \boldsymbol{t} = h(\boldsymbol{M}) + \boldsymbol{e} = (& \underbrace{0}_{b_0^{(0)}} \cdot \underbrace{1110001001_2}_{\boldsymbol{s}^{(0)}} + \\ & \underbrace{0}_{b_1^{(0)}} \cdot \underbrace{0111000100_2}_{\boldsymbol{s}^{(1)}} + \\ & \underbrace{1}_{b_2^{(0)}} \cdot \underbrace{1011100010_2}_{\boldsymbol{s}^{(2)}} + \\ & \cdots) + \underbrace{\ldots 010_2}_{\boldsymbol{e}} \end{aligned}$$

Based on the feedback function $u_{i+10} = u_{i+7} + u_i$, let $\boldsymbol{\Delta} = 10010000001_2$. Since $|\boldsymbol{\Delta}| = n + 1 = 11 \leq k_1 + k_2 + 1 = 15$, the adversary may set $\mathring{\boldsymbol{\Delta}} = \text{Shift}^{k_1+k_2+1}(\boldsymbol{\Delta}, j)$ for $j = 0, \ldots, k_1 + k_2 - n = 4$ and, using the attack of Section 4.2.1, learn the following 5 linear equations in the bits of $\boldsymbol{x}, \boldsymbol{y}$:

$$\begin{aligned} y_0 + y_7 + x_0 &= 1 && (10) \\ y_1 + y_8 + x_1 &= 0 && (11) \\ y_2 + y_9 + x_2 &= 1 && (12) \\ y_3 + x_0 + x_3 &= 0 && (13) \\ y_4 + x_1 &= 0 && (14) \end{aligned}$$

The adversary needs $k_1 + k_2 = 14$ linearly independent linear equations in total, and he can get them by using additional MIM patterns. For example, let

$$\begin{aligned} \boldsymbol{w} = \mathrm{Shift}^{2n+1}(\boldsymbol{\Delta}, n) + \mathrm{Shift}^{2n+1}(\boldsymbol{\Delta}, 0) &= 100100000010000000000_2 + 000000000010010000001_2 \\ &= 100100000000010000001_2 \end{aligned}$$

Since $|\boldsymbol{w}| = 2n+1 = 21 \leq 30 = 2(k_1+k_2+1)$ and $2|r = 40$, the adversary may use $\mathring{\boldsymbol{w}} = \mathrm{Shift}^{2(k_1+k_2+1)}(\boldsymbol{w}, j)$ with $j = 0, \ldots, 9$ in the 2-round version of the attack of Section 4.2.1 or 4.2.2. For example, with $j = 0$, he uses:

$$\mathring{\boldsymbol{w}} = \mathrm{Shift}^{2(k_1+k_2+1)}(\boldsymbol{w}, 0) = \underbrace{\underbrace{0}_{\bar{z}^{(1)}}\ \underbrace{0000}_{\bar{\boldsymbol{a}}^{(1)}}\ \underbrace{0000100100}_{\bar{\boldsymbol{b}}^{(1)}}}_{\mathring{\boldsymbol{w}}^{(1)}}\ \underbrace{\underbrace{0}_{\bar{z}^{(0)}}\ \underbrace{0000}_{\bar{\boldsymbol{a}}^{(0)}}\ \underbrace{0010000001_2}_{\bar{\boldsymbol{b}}^{(0)}}}_{\mathring{\boldsymbol{w}}^{(0)}}$$

Since both $\bar{\boldsymbol{a}}^{(0)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(0)}\boldsymbol{y} + \bar{z}^{(0)} = 0$ and $\bar{\boldsymbol{a}}^{(1)}\boldsymbol{x} + \bar{\boldsymbol{b}}^{(1)}\boldsymbol{y} + \bar{z}^{(1)} = 0$, the identification session results in acceptance with overwhelming probability, yielding 2 linear equations:

$$\begin{aligned} j = 0, \quad y_0 + y_7 &= 0 && (15) \\ y_2 + y_5 &= 0 && (16) \end{aligned}$$

Continuing with $j = 1, \ldots, 9$, the adversary gets the following 4 additional linear equations (note that no equations are generated for $j = 1$ or $j \geq 4$):

$$\begin{aligned} j = 2, \quad y_2 + y_9 &= 0 && (17) \\ y_4 + y_7 &= 0 && (18) \\ j = 3, \quad y_3 + x_0 &= 0 && (19) \\ y_5 + y_8 &= 0 && (20) \end{aligned}$$

At this point the adversary has 11 linearly independent linear equations (10-20) in the $k_1 + k_2 = 14$ bits of $\boldsymbol{x}$ and $\boldsymbol{y}$ and can get the remaining three equations, for example, by trying some other MIM pattern. Due to space limitations (the matrix U has $m = 600$ columns), we also omit the detailed description of the recovery of $\boldsymbol{s}$, assuming it should already be pretty clear from Section 4.2.3.

4.3 A Passive Attack

When p and $\boldsymbol{e}$ are secret, the signature scheme used in Trusted-HB is *ε-otp-secure*. Based on this proven fact, Theorem 1 of [15] states that any MIM attack on Trusted-HB has a probability of success of at most ε because the reader's view of the noise is unknown to the adversary. In fact, this claim is incorrect since $\boldsymbol{e}$ is a recycled version of $\boldsymbol{v}$, and resembles the situation in which someone re-uses a one-time pad, in which all bets are off. Consequently, there is no reason why Trusted-HB should be secure even against passive attacks. While the attack that we are about to describe is based on the specific extraction procedure recommended by the developers of Trusted-HB, it simply highlights the fundamental flaw in the security proof of the scheme.

The von Neumann randomness extraction rule, described in (5), states that if $v^{(0)} \neq v^{(1)}$, then $e_0 = v^{(0)}$. Since at least one of $(v^{(0)} + v^{(1)}) = 0$ and $(v^{(0)} + e_0) = 0$ must be true, this can be expressed by the following equation which is always true:

$$(v^{(0)} + v^{(1)})(v^{(0)} + e_0) = 0 \tag{21}$$

Denote the first row of the Toeplitz matrix U by $\boldsymbol{w} = \overline{u_{m-1} \dots u_0}$. Then we can re-write (21) as a quadratic equation in the bits of $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{w}$:

$$\begin{aligned} 0 &= (\boldsymbol{a}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)}\boldsymbol{y} + z^{(0)} + \boldsymbol{a}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\boldsymbol{y} + z^{(1)})(\boldsymbol{a}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)}\boldsymbol{y} + z^{(0)} + t_0 + \boldsymbol{M}\boldsymbol{w}) \\ &= \underbrace{\boldsymbol{a}^{(1)}\boldsymbol{x} \cdot \boldsymbol{a}^{(0)}\boldsymbol{x} + \boldsymbol{a}^{(0)}\boldsymbol{x} \cdot (z^{(1)} + t_0 + 1) + \boldsymbol{a}^{(1)}\boldsymbol{x} \cdot (z^{(0)} + t_0)}_{\leq \frac{k_1(k_1+1)}{2} \text{ monomials}} \\ &\quad + \underbrace{\boldsymbol{b}^{(1)}\boldsymbol{y} \cdot \boldsymbol{b}^{(0)}\boldsymbol{y} + \boldsymbol{b}^{(0)}\boldsymbol{y} \cdot (z^{(1)} + t_0 + 1) + \boldsymbol{b}^{(1)}\boldsymbol{y} \cdot (z^{(0)} + t_0)}_{\leq \frac{k_2(k_2+1)}{2} \text{ monomials}} \\ &\quad + \underbrace{(\boldsymbol{a}^{(1)}\boldsymbol{x}) \cdot (\boldsymbol{b}^{(0)}\boldsymbol{y}) + (\boldsymbol{a}^{(0)}\boldsymbol{x}) \cdot (\boldsymbol{b}^{(1)}\boldsymbol{y})}_{\leq k_1 k_2 \text{ monomials}} \\ &\quad + \underbrace{((\boldsymbol{a}^{(0)} + \boldsymbol{a}^{(1)})\boldsymbol{x} + (\boldsymbol{b}^{(0)} + \boldsymbol{b}^{(1)})\boldsymbol{y} + (z^{(0)} + z^{(1)})) \cdot (\boldsymbol{M}\boldsymbol{w})}_{\leq (k_1+k_2+1)m \text{ monomials}} \\ &\quad + (z^{(0)} + z^{(1)})(z^{(0)} + t_0) \end{aligned} \tag{22}$$

where $\boldsymbol{a}^{(0)}, \boldsymbol{b}^{(0)}, z^{(0)}, \boldsymbol{a}^{(1)}, \boldsymbol{b}^{(1)}, z^{(1)}$, and $\boldsymbol{M}$ are known.

All the monomials that appear in (22) are of degree at most 2, and their number is

$$Q \leq k_1(k_1 + 1)/2 + k_2(k_2 + 1)/2 + k_1k_2 + (k_1 + k_2 + 1)m = O(k_2^2 r)$$

assuming $k_2 > k_1$. If the adversary passively observes $O(Q)$ identification sessions, he can get Q linearly independent linear equations in all these monomials and solve the system for $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}$ by linearization in time $O(Q^3) = O((k_2^2 r)^3)$. Note that the knowledge of $\boldsymbol{w} = \overline{u_{m-1} \dots u_0}$ gives us $u_0, \dots, u_{n-1}$, and since the sequence $u_{1-n}, \dots, u_0, \dots$ is generated by an LFSR, the adversary can express every bit of $\boldsymbol{s} = \overline{u_0 \dots u_{1-n}}$ as a linear combination of $u_0, \dots, u_{n-1}$. However, the complexity of this straightforward implementation of the attack is high: the query complexity in terms of the number of sessions is $Q = O(k_2^2 r)$, the time complexity is $O(Q^3) = O((k_2^2 r)^3)$, and the memory complexity, dominated by the storage needed for the system of equations, is $O(Q^2) = O((k_2^2 r)^2)$. For the recommended parameter values $k_2 = 512 = 2^9, r = 1164 \approx 2^{10}$, this attack requires $O(2^{28})$ queries, time of $O(2^{84})$, and memory of $O(2^{56})$, which makes it infeasible in practice.

As explained in 4.1, the feedback polynomial is usually known, and in this case (22) can be greatly simplified, leading to a much lower complexity. Each bit of $\boldsymbol{w}$ can be expressed as a known linear combination of the bits of $\boldsymbol{s}$, so given $\boldsymbol{M}$, one can compute $\boldsymbol{M}'$ of length n such that $\boldsymbol{M}\boldsymbol{w} = \boldsymbol{M}'\boldsymbol{s}$. Thus, (22) becomes:

$$0 = (\boldsymbol{a}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)}\boldsymbol{y} + z^{(0)} + \boldsymbol{a}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\boldsymbol{y} + z^{(1)})(\boldsymbol{a}^{(0)}\boldsymbol{x} + \boldsymbol{b}^{(0)}\boldsymbol{y} + z^{(0)} + t_0 + \boldsymbol{M}'\boldsymbol{s}) \tag{23}$$

The number of monomials of degree ≤ 2 is

$$Q' \leq k_1(k_1+1)/2 + k_2(k_2+1)/2 + k_1k_2 + (k_1+k_2+1)n = O(k_2^2) \tag{24}$$

assuming $k_2 > k_1, n$.

The complete algorithm is as follows:

- Observe $O(Q')$ sessions and convert each equation of type (22) into a quadratic equation in the bits of $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$. (To simplify checking for linear independence, this step can be combined with the next step).

  To produce the $\boldsymbol{M}'$ of length n such that $\boldsymbol{M}'\boldsymbol{s} = \boldsymbol{M}\boldsymbol{w}$, we use the MIM pattern $\boldsymbol{\Delta}$ of (7). Let

  $$\begin{aligned} \boldsymbol{M}'' &= \text{Shift}^{m+n-1}(\boldsymbol{M}, n-1) \\ \boldsymbol{w}'' &= \boldsymbol{w}\|\overline{u_{-1}\ldots u_{1-n}} = \overline{u_{m-1}\ldots u_{1-n}} \end{aligned}$$

  Thus, $\boldsymbol{M}\boldsymbol{w} = \boldsymbol{M}''\boldsymbol{w}''$. Since $\boldsymbol{\Delta}$ is a complete MIM pattern,

  $$\forall \boldsymbol{A} \in \{0,1\}^{m+n-1}, j = 0, \ldots, m-2, \quad \boldsymbol{A}\boldsymbol{w}'' = (\boldsymbol{A} + \text{Shift}^{m+n-1}(\boldsymbol{\Delta}, j)) \cdot \boldsymbol{w}''$$

  Therefore, we can obtain the $\boldsymbol{M}'$ by the following procedure:

  ```
  for $j \leftarrow m+n-1$ downto $n$ do
      if $w''_j = 1$ then
          set $\boldsymbol{M}'' \leftarrow \boldsymbol{M}'' + \text{Shift}^{m+n-1}(\boldsymbol{\Delta}, j-n)$;    /* force $M''_j = 0$ */
      end
  end
  Set $\boldsymbol{M}' \leftarrow \overline{M''_{n-1}\ldots M''_0}$;    /* $\boldsymbol{M}'' = \underbrace{0\ldots0}_{m-1}\|\boldsymbol{M}'$ */
  ```

  The output of the algorithm satisfies $\boldsymbol{M}'\boldsymbol{s} = \boldsymbol{M}' \cdot \overline{u_0 \ldots u_{1-n}} = \boldsymbol{M}''\boldsymbol{w}'' = \boldsymbol{M}\boldsymbol{w}$. The computation requires minimal extra memory and is performed in time $O(ml)$.

- Solve for $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}$ by linearization in time $O(Q'^3) = O(k_2^6)$.

Thus, the query complexity is $O(k_2^2)$ and the time complexity is $O(k_2^6)$. Memory is used mainly for two purposes: storing transcripts $\boldsymbol{M}$ one at a time (m bits) and storing $O(Q')$ precomputed equations (23) to solve the system ($O(Q'^2)$ bits), so the total memory complexity is $O(m + Q'^2) = O(k_2^4)$. Note that the complexity of the attack does not depend on η. While the query complexity is pretty small, if fewer than Q' queries are available, it could still be possible to solve the quadratic system reasonably fast using the methods described in [33].

Consider the concrete values $k_1 = 80, k_2 = 512, n = 101$ recommended for Trusted-HB. The query complexity is $Q' = 235320 = O(2^{18}) = O(k_2^2)$, so the total time complexity is $O(2^{54})$ and the memory complexity is $O(2^{36})$, which are (barely) feasible.

4.3.1 A Toy Example

Consider the following toy example with unrealistically small parameter values: $k_1 = 1, k_2 = 1, r = 4, n = 2$, so $m = (k_1 + k_2 + 1)r = 12$. The number of rounds r is artificially small to keep the transcript short, so we will only consider those sessions where the von Neumann procedure succeeds in producing at least one bit of output needed for the passive attack; if only a single bit is produced, we denote the other bit by '?' (the adversary does not look at it anyway). Let the connection polynomial of the LFSR be $p(X) = X^2 + X + 1$, which corresponds to $u_{i+2} = u_{i+1} + u_i$ for $-1 = 1 - n \leq i \leq m - 1 = r(k_1 + k_2 + 1) - 1 = 11$ and $\mathbf{\Delta} = 111_2$. Let the secrets be $\boldsymbol{x} = 1, \boldsymbol{y} = 0, \boldsymbol{s} = 01_2$, so the Toeplitz matrix is

$$\begin{aligned} U &= \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} \\ u_{-1} & u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \end{aligned} \tag{25}$$

According to (24), we need about $k_1(k_1 + 1)/2 + k_2(k_2 + 1)/2 + k_1k_2 + (k_1 + k_2 + 1)n = 9$ identification sessions to recover the secrets. The transcripts of the eavesdropped identification sessions are summarized in Table 2. For example,

$$\underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\boldsymbol{t}^{(0)}} = \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}}_{U} \cdot \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\boldsymbol{M}^{(0)}} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\boldsymbol{e}^{(0)}}$$

Using the assumed knowledge of p, the adversary can compute:

$$\boldsymbol{w} = \begin{bmatrix} s_1 \\ s_1 + s_0 \\ s_0 \\ s_1 \\ s_1 + s_0 \\ s_0 \\ s_1 \\ s_1 + s_0 \\ s_0 \\ s_1 \\ s_1 + s_0 \\ s_0 \end{bmatrix}$$

Since $|\boldsymbol{x}| = |\boldsymbol{y}| = 1$, let $x = x_0, y = y_0$ and write a system of equations of type (23), one per eavesdropped

Table 2: Transcript Summaries (in binary)

Session j	Round i	$(\boldsymbol{b}^{(i)})^{(j)}$	$(\boldsymbol{a}^{(i)})^{(j)}$	$(z^{(i)})^{(j)}$	$((\nu^{(i)})^{(j)})$	$(\boldsymbol{e}^{(j)})$	$\boldsymbol{t}^{(j)} = U\boldsymbol{M}^{(j)} + \boldsymbol{e}^{(j)}$
0	0	0	1	0	(1)	$\left(\begin{bmatrix}1\\0\end{bmatrix}\right)$	$\begin{bmatrix}1\\1\end{bmatrix}$
	1	0	0	0	(0)		
	2	1	0	0	(0)		
	3	0	1	0	(1)		
1	0	1	0	1	(1)	$\left(\begin{bmatrix}1\\?\end{bmatrix}\right)$	$\begin{bmatrix}0\\?\end{bmatrix}$
	1	1	1	0	(1)		
	2	1	0	1	(1)		
	3	0	1	1	(0)		
2	0	1	0	0	(0)	$\left(\begin{bmatrix}0\\0\end{bmatrix}\right)$	$\begin{bmatrix}0\\1\end{bmatrix}$
	1	1	1	0	(1)		
	2	0	0	0	(0)		
	3	0	0	1	(1)		
3	0	0	1	0	(1)	$\left(\begin{bmatrix}1\\0\end{bmatrix}\right)$	$\begin{bmatrix}1\\0\end{bmatrix}$
	1	1	1	1	(0)		
	2	0	1	1	(0)		
	3	1	1	0	(1)		
4	0	1	0	1	(1)	$\left(\begin{bmatrix}1\\?\end{bmatrix}\right)$	$\begin{bmatrix}0\\?\end{bmatrix}$
	1	0	0	1	(1)		
	2	0	0	1	(1)		
	3	0	1	1	(0)		
5	0	1	0	1	(1)	$\left(\begin{bmatrix}1\\?\end{bmatrix}\right)$	$\begin{bmatrix}0\\?\end{bmatrix}$
	1	1	1	1	(0)		
	2	0	1	0	(1)		
	3	1	0	1	(1)		
6	0	0	0	1	(1)	$\left(\begin{bmatrix}0\\?\end{bmatrix}\right)$	$\begin{bmatrix}1\\?\end{bmatrix}$
	1	1	1	0	(1)		
	2	0	1	1	(0)		
	3	0	0	1	(1)		
7	0	1	0	0	(0)	$\left(\begin{bmatrix}0\\1\end{bmatrix}\right)$	$\begin{bmatrix}0\\1\end{bmatrix}$
	1	0	1	0	(1)		
	2	1	0	1	(1)		
	3	0	1	1	(0)		
8	0	0	0	1	(1)	$\left(\begin{bmatrix}1\\?\end{bmatrix}\right)$	$\begin{bmatrix}0\\?\end{bmatrix}$
	1	0	0	0	(0)		
	2	0	0	0	(0)		
	3	1	1	1	(0)		

session:

$$\begin{cases} x(x+1+s_1)=0 \\ (x+1)(y+1+s_0+s_1)=0 \\ x(y+s_1)=0 \\ (y+1)(x+1)=0 \\ y(y+1+s_0)=0 \\ x(y+1+s_0+s_1)=0 \\ (x+y+1)(s_0+s_1)=0 \\ (y+x)y=0 \\ 1+s_0=0 \end{cases} \Leftrightarrow \begin{bmatrix} 0&0&0&0&0&0&1&0&0 \\ 1&1&1&1&1&1&1&0&0 \\ 0&0&0&0&1&0&1&0&0 \\ 1&1&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 1&0&0&0&1&1&1&0&0 \\ 0&0&1&1&0&1&1&1&1 \\ 0&1&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ s_0 \\ s_1 \\ xy \\ xs_0 \\ xs_1 \\ ys_0 \\ ys_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The system contains 9 linearly independent linear equations in 9 monomials and, thus, can be easily solved for x, y, s_0, s_1 by Gaussian elimination.

5 Conclusion

In this paper, we described several attacks on Trusted-HB in the realistic scenario when the connection polynomial of the LFSR is known, described a potential problem with generating a sufficiently long random-looking bitstring for low noise rates, and showed why Trusted-HB cannot be trusted even when the connection polynomial is secret. However, the complexity of our attack is relatively high, and finding a more efficient attack in this case is left as an open problem. More generally, the problem of finding lightweight LPN-based identification schemes that are provably secure against arbitrary man-in-the-middle attacks remains an open direction for research.

References

[1] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg. On the inherent intractability of certain coding problems. In *IEEE Transactions on Information Theory*, volume IT-24, pages 384–386, May 1978.

[2] Johan Håstad. Some optimal inapproximability results. In *STOC*, pages 1–10, 1997.

[3] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66.

[4] Arie Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308.

[5] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the HB and HB+ protocols. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 73–87.

[6] Jonathan Katz and Adam Smith. Analyzing the HB and HB+ protocols in the ”large error” case. Technical Report 2006/326, Cryptology ePrint Archive.

[7] Jonathan Katz. Efficient cryptographic protocols based on the hardness of learning parity with noise. In *IMA Int. Conf.*, pages 1–15, 2007.

[8] Henri Gilbert, Matthew Robshaw, and Hervé Silbert. An active attack against HB+ – a provable secure lightweight authentication protocol. Technical Report 2005/237, Cryptology ePrint Archive.

[9] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. 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.

[10] Selwyn Piramuthu. HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In *CollECTeR Europe Conference*, June 2006.

[11] J. Munilla and A. Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51:2262–2267, June 2007.

[12] Xuefei Leng, K. Mayes, and K. Markantonakis. HB-MP+ protocol: An improvement on the HB-MP protocol. *RFID, 2008 IEEE International Conference on*, pages 118–124, April 2008.

[13] Dang Nguyen Duc and Kwangjo Kim. Securing HB+ against GRS man-in-the-middle attack. In *Proceedings of SCIS 2007, Abstracts*, page 123.

[14] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. $\mathrm{HB}^{\#}$: Increasing the security and efficiency of HB^{+}. In *EUROCRYPT*, pages 361–378, 2008.

[15] Julien Bringer and Hervé Chabanne. Trusted-HB: A low-cost version of HB $^{+}$ secure against man-in-the-middle attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, 2008.

[16] Ghaith Hammouri, Erdinç Öztürk, Berk Birand, and Berk Sunar. Unclonable lightweight authentication scheme. In *ICICS*, volume 5308 of *Lecture Notes in Computer Science*, pages 33–48, 2008.

[17] H. Gilbert, M.J.B. Robshaw, and Y. Seurin. Good variants of HB+ are hard to find. In *Financial Crypto*, 2008.

[18] K. Ouafi, R. Overbeck, and S. Vaudenay. On the security of HB# against a man-in-the-middle attack. Accepted at ASIACRYPT 2008.

[19] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[20] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM*, pages 378–389, 2005.

[21] Marc P. C. Fossorier, Miodrag J. Mihaljević, Hideki Imai, Yang Cui, and Kanta Matsuura. An algorithm for solving the LPN problem and its application to security evaluation of the HB protocols for RFID authentication. In *Progress in Cryptology - INDOCRYPT 2006*, volume 4329 of *Lecture Notes in Computer Science*, pages 48–62.

[22] Éric Levieil and Pierre-Alain Fouque. An improved LPN algorithm. In *Security and Cryptography for Networks*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359, 2006.

[23] Zbigniew Golebiewski, Krzysztof Majcher, Filip Zagórski, and Marcin Zawada. Practical attacks on HB and HB+ protocols. Accepted at Inscrypt 2008.

[24] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17:281–308, 1988.

[25] Adi Shamir. SQUASH - a new MAC with provable security properties for highly constrained devices such as RFID tags. In *FSE*, pages 144–157, 2008.

[26] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[27] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 235–243, 1990.

[28] Hugo Krawczyk. LFSR-based hashing and authentication. In *CRYPTO*, pages 129–139, 1994.

[29] John von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards*, 12:36–38, 1951.

[30] K. Nohl and H. Plötz. Mifare - little security despite obscurity. Talk at 24C3, available at `http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html`.

[31] Yuval Peres. Iterating Von Neumann's procedure for extracting random bits. *The Annals of Statistics*, 20(1):590–597, March 1992.

[32] Antoine Joux and Michel Mitton. Fast correlation attacks: an algorithmic point of view. In *EUROCRYPT 2002, LNCS 2332*, pages 209–221, 2002.

[33] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT*, pages 392–407, 2000.