Pitfalls in Ultralightweight Authentication Protocol Designs

Gildas Avoine, Xavier Carpent, Julio Hernandez-Castro

Abstract—This article introduces prudent engineering practices and offers recommendations to follow, together with typical mistakes to avoid, when designing new ultralightweight authentication protocols. This work can help, as a sanity check, designers of RFID, NFC and sensor networks based security solutions to improve the security, reliability and longevity of ultralightweight authentication protocol designs. Additionally, it aims to help reviewers to quickly distinguish what is really new and worthy in a research area that has been flooded lately with proposals of dubious quality.

Index Terms—Ultralightweight Protocols, Authentication, Cryptanalysis, Security, Privacy, RFID, NFC, Sensor networks

1 INTRODUCTION

RFID authentication is a very active and challenging research topic. It has strong requirements such as security of course, but also privacy, and yet very strong constraints such as powerful adversaries and very constrained devices. This last point in particular is the focus of a significant part of the research community, and has spawned hundreds of papers [8]. Indeed, there is a strong incentive to build RFID tags below 5 cents that still provide some level of security. Such tags can not contain a battery, must make do with very few logic gates, and must be able to complete an authentication in very little time.

In response to that, a number of protocols have been created that rely on extremely simple operations to do basic cryptography. Such protocols have been named “ultralightweight”, a term coined by Chien in [27]. Although there is no precise definition adopted by authors of ultralightweight protocols, they can be broadly defined as protocols that are aimed at requiring about 1.5K GE (gate equivalents) or less, relying on only very simple operations on the tag’s side, and willing to accept compromises to offer some security level (See Chien’s definition [27]). This includes, in particular, most of the authentication protocols aimed at passive and inexpensive RFID tags.

As one might expect, most of these protocols are usually very weak, and are typically broken very quickly. More importantly, there tends to be a certain lack of originality, novelty, and scientific soundness in these proposals, which is counter-productive and contributes to the relative stagnation in the field. A plausible reason why these papers are accepted is that reviewers not familiar with the related literature fail to notice common mistakes in these protocols.

In this paper, we give a mature perspective on the situation in the field of ultralightweight authentication protocols, and provide a description of most of the typical flaws that frequently undermine new proposals. This may be useful for potential reviewers to quickly determine if a proposal is worthy of publication or could be broken easily. It could also be used as a sanity check for ultralightweight protocol designers. Some authors, notably D’Arco and De Santis, highlighted in [29] a set of slightly different but related and often complementary observations to the ones we will examine in further detail and expand on this work.

The structure of the article is the following. In Section 2 we give some statistics related to the field and give additional motivations for the article. In Section 3 we give a very quick description of the attacker model in ultralightweight authentication. In Section 4 we present the most typical mistakes found in ultralightweight protocols. In Section 5 we mention the problematic of dubious security proofs and detail why they are usually flawed. In Section 6 we present some recent proposals and propose attacks and comments on them based on the discussions from Sections 4 and 5. In Section 7 we end on a more positive note by presenting what we believe are the more promising approaches, and we finally present our conclusions in Section 8.

2 MOTIVATIONS

This section presents a few facts on the state of the art of ultralightweight protocols.

Table 1 presents a list of most prominent ultralightweight authentication protocols, and the first attack on each of them, when applicable. It also shows the date of publication of each article and highlights the time in months elapsed between the publication of a protocol and the publication of an attack on it. What we consider to be a “first attack” is the chronologically first traceability or disclosure attack that targets the specific construction of a protocol (we discarded desynchronization attacks because, although important, they do not target specific protocol weaknesses but rather general issues in their structure). The dates, both for protocols and attacks, are taken to be the earliest after becoming public (e.g., eprint or date of conference over proceedings).

Figure 1 shows the distribution of the time required for the publication of an attack, based on data from Table 1. Although there are many factors influencing the data, it is clear that the time for publication of a serious attack is extremely short. The average number of months required is about 9.5. Half of the protocols are broken within 8 months.

1. It is relevant to point out that gate counts differ in the literature, sometimes by a large margin. See a discussion on the subject in [82], [85].

2. We discarded Gossamer in these calculations since no attack has been published on it yet to the best of our knowledge. There are, however, existing desynchronization attacks (see [125], [110], and [121]), but they are related to the structure of Gossamer, not its operations. These papers propose different fixes to the problem.
and most of them within a year.

Note that depending on the journal or conference, it takes significant time to publish an attack on a protocol. We estimate that at least three to four months are required from finished research to publication. Generally, it might also take time for the proceedings to become available. Note finally that some of these protocols have been published in relatively low-visibility venues, which has definitely helped increase their life expectancy. It is probably safe to assume that on average, a typical ultralightweight protocol is broken in under four months following its publication.

In most cases, these protocols repeat time and again the same typical mistakes, and no progress is achieved. Many proposals present striking similarities with existing models; there are different experiments. The most widely used models are due to Juels and Weis [65], Vaudenay [117], which treat active attackers as being more powerful than eavesdroppers, despite relay attacks being in fact not harder than eavesdropping (see for instance the library [78]).

3 Generic Attack Model

Ultralightweight authentication protocols must primarily ensure that neither an impersonation attack, nor a key-recovery attack (even partial-recovery) is feasible. Protocol analyses usually consider the Dolev-Yao model [33], namely a model where the adversary is powerful but not almighty: the adversary cannot guess random numbers chosen in a sufficiently large space, she cannot decrypt or create valid ciphertexts without the correct secret, and she cannot retrieve private keys from public information. However, she can easily communicate with both a tag or a reader, eavesdrop communications in both directions, or perform relay attacks (possibly modifying the messages). However, she can easily communicate with both a tag or a reader, eavesdrop communications in both directions, or perform relay attacks (possibly modifying the messages).

In the following, we present briefly the attack model, and then go through the typical flaws that commonly appear in these ultralightweight protocols.

Figure 1. Histogram of the time for ultralightweight protocols to be broken.
Deursen, Mauw, and Radomirovic [115], Hermans, Pashalidis, Vercauteren, and Preneel [55], and Avoine, Coisel, and Martin [12]. It is today strongly advised to use such a well-established model instead of ad-hoc ones. A comparison of existing models was done by Coisel and Martin [25].

Designing a privacy-friendly authentication protocol is a difficult question. While a classical challenge-response protocol (where the identity of the prover is not sent in the clear on the channel) seems to be privacy-friendly at first sight, it suffers from a linear complexity search (all the keys stored by the verifier must be checked to identify the correct prover) and does not ensure forward-privacy: if the prover is compromised, the privacy of the past interactions is no longer ensured. The latter issue can be mitigated by updating the key, but desynchronization attacks must be prevented in such a case. Vaudenay actually demonstrated that only public-key cryptography can ensure the highest attainable privacy level [115]. Ultra-lightweight authentication protocols do not necessarily target the highest privacy level, though. Avoine, Coisel, and Martin’s model of privacy [12] for instance introduced the notion of existential and universal traceability (intuitively, a protocol is existentially traceable if it is possible to identify a tag from two communication traces, unless a legitimate authentication took place between them).

4 COMMON FLAWS

4.1 Linearity and T-functions

We use here the same definition of T-functions proposed by Klimov and Shamir in [69], which is the following: A T-function is a mapping from n-bit words to n-bit words in which for each 0 ≤ i < n, the bit i in any output word depends only on bits 0, 1, ..., i of any input word. This concept is very useful because all the Boolean operations and most of the numeric operations in modern processors are T-functions. Additionally, the composition of T-functions results also in a T-function.

These have many nice properties, but almost by definition also a quite undesirable one: they achieve very poor levels of diffusion. By definition, in a T-function it is not possible that all output bits depend on all input bits, which is the ideal scenario for security purposes. This is particularly dangerous in cryptographic applications, lightweight or otherwise. The only reasonable way to address this shortcoming is by combining these operations with other which do not exhibit this characteristic. But unfortunately many designers do not follow this simple combination rule, and have proposed schemes entirely based on T-functions which are doomed to fail.

We start by showing some common mistakes found in the UMAP family of protocols. These were pioneer proposals in many ways but they incurred in what, with the hindsight gained after more than 8 years of research in the area, now looks like quite elementary mistakes. We show in the following the scheme used in LMAP, presented in [94] and pictured in Figure 2, which is still one of the most clear examples of this vulnerability.

As we can see in Figure 2 this early protocol is composed exclusively of T-functions (modular addition, ⊕, ∨). This seriously limits its security due to their poor diffusion characteristics. Not only the message generation is marred by this, but it also affects the key update phase. This same

![Figure 2. The LMAP protocol.](http://www.ieee.org/publications_standards/publications/rights/index.html for more information.)
almost exclusively in the use of Cyclic Redundancy Codes (CRCs) that many authors seem to forget are also linear and, as such, offer very little security if at all.

This is excusable to a certain extend in proposals that want to be compliant with the EPC-C1-G2 standard recommendations, but then the security claims of the proposed protocols should be adjusted down accordingly. CRCs are linear and they should under no circumstances be employed in cryptography. For a further example of this, see Sect. 5.2

As an additional example of how pervasive this mistake is, the papers [42] and [91] are good very recent illustrations of this common failure. Not surprisingly they were both quickly dismantled in [2].

### 4.2 Biased output

Another important weakness of many lightweight schemes proposed in the last years is that some of the operations used have a biased output, a feature that in many cases quickly dismantled in [2].

#### 4.2.1 Another important weakness of many lightweight schemes

For example, if \( x \lor y = 0 \), then \( x = y = 0 \), which discloses both the inputs. This happens roughly 25% of the times (similarly with AND, of course). This could be more than enough to, after seeing some exchanges, be able to completely recover all the inputs. An additional undesirable property derived from these highly biased outputs is the possibility of message insertion. For instance, imagine that one of the exchanged messages, used for synchronization or authentication purposes, is the result of one of this biased operators. Then, an attacker can easily impersonate it with a high probability, so it is a very bad idea to use it for authentication purposes, as proposed in many protocols. Let’s exemplify again over the LMAP protocol. We have message \( B = (IDS \lor K_2) + n_1 \), a typical construction that has been seen in this or similar forms many times later. The issue here is that \( B \) is very biased and thus can be impersonated with high probability even for an attacker who doesn’t know any of the secrets \( K_2 \) or \( n_1 \). Even more worryingly, as the value of \( B \) is public (it is exchanged in the open), the attacker can use \( B - T \) as a very good approximation to the secret \( n_1 \) (on average, 75% of the bits are correct), and this approximation can be used later in other parts of the protocol to approximate the secret.

### 4.3 Rotations

Rotations have been used for a long time in cryptography. Modern block ciphers and hash functions still mostly rely on ARX (addition, rotation, XOR) designs. Rotations are extremely cheap to implement, and they bring diffusion, which complements nicely the addition and the XOR (which exhibit poor diffusion properties, as shown in Sect. 4.1.

Fixed-amount rotations are typically used in ARX designs, but data-dependent rotations, as first featured in the RC5 block cipher [103], also exist.

The SASI [27] protocol was the first ultralightweight authentication protocol to feature data-dependent rotations as far as we know. Since then, most ultralightweight protocols have used them, and in many cases they are the weak spot for ad-hoc attacks. Two types of rotations are typically used in these protocols: modular rotations (as used in RC5) and Hamming weight rotations. In the following, we denote the rotation operation by \( \text{Rot}(x, y) \), in accordance with the literature. The rotation consists in a circular left shift of \( x \) by \( r(y) \) positions, where \( r(y) = y \mod L \) for modular rotations, and \( r(y) = H(y) \) for Hamming weight rotations.

One thing to bear in mind is that a rotation is a permutation and is therefore linear, inheriting all the pitfalls commented in Sect. 4.1.

The most important shortcoming with data-dependent rotations is that there are only \( L \) possible outputs (where \( L \) is the size in bits of the input). Moreover, the distribution of these outputs is known a priori. The modulo operation has a uniform distribution over uniform input, and the Hamming weight has binomial distribution \( B(L, 1/2) \) over uniform input. Figure 3 shows the distribution of probability of \( r(y) \) for the two rotations, with \( y \) a uniform random variable and with \( L = 96 \).

Modular rotations have a maximal entropy (\( \log_2 L = 6.58 \) bits), since each shift is equiprobable. This minimizes the advantage of the adversary for guessing the output. Nevertheless, they are quite probable to behave like the identity operation (probability of \( 1/L \)). This has led to many attacks, whether when the attacker assumes this blindly (as the desynchronization attack on PUMAP in [9]) or when she is able to verify it (as the traceability attack on modular SASI in [56]).

Hamming weight rotations have worse entropy (e.g. 4.34 bits in the case of \( L = 96 \)) than modular rotations. In particular, the three-sigma rule implies that, for instance in the case of \( L = 96 \), the number of bits rotated is between 33 and 63 in 99.7% of cases. It is therefore even easier for an attacker to guess the output of a rotation. However, Hamming weight rotations virtually never behave like the identity. Indeed, this would require either \( r(y) = 0 \) or \( r(y) = L \), which happens only when \( y = 0 \) or \( y = 2^L - 1 \), respectively.

Mod \( n \) cryptanalysis [67] is also a promising tool to attack schemes using rotations and additions, although it has never been applied in the cryptanalysis of an ultralightweight protocol. It has, on the other hand, been used to successfully attack block ciphers such as RCSP and M6, which use the same kind of operations.

### 4.4 Message Composition

Securely designing the messages exchanged over an ultralightweight protocol is a difficult and open problem. Keeping the secrets exchanged as secure as possible against any other parts of the protocol to approximate the secret.
leakage is indeed a big challenge, particularly in such constrained environments. Generally speaking, the messages should guarantee good confusion and diffusion properties for the secrets. That is, the secret should be thoroughly involved in the construction of the messages, and a subtle change in the secret should result in completely different messages. In block ciphers, for instance, these two properties are usually achieved by using iterated substitution-permutation networks. However, due to the constraints of ultralightweight protocols, messages are usually built using a handful of operations, and in many cases good confusion and diffusion levels are not satisfied.

In LMAP for instance, the key update phase is defined by

\[ IDS^{(n+1)} = (IDS^{(n)} + (n_2^{(n)} \oplus K_4^{(n)})) \oplus ID, \]  

(1)

where we can see that the ID, a secret that the protocol is designed to protect, is simply XORed with a mixture of public and secret values. This operation clearly exhibits poor confusion and diffusion properties. This quite frequent feature heuristically leads to major leakage of secret bits, as the rest of the message the ID is combined with may include bias, or be partially known by the adversary.

There are many more examples that exhibit transgressions to this rule of thumb for minimizing secret leakage, even within the most recent proposals. Consider for instance this message of the RAPP protocol [111]:

\[ C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID, \]  

(2)

where we can see again that the ID is at the out most position of the message construction. This would have no serious consequences if the rest of the message was perfectly random and unknown to an adversary (as in a One Time Pad), but this is not the case here.

4.5 Knowledge Accumulation

4.5.1 Knowledge Accumulation of a Static Secret

If partial leakage of a static secret occurs in a round of a protocol, there is an obvious traceability issue. Indeed, it becomes possible for an attacker to correlate two leaked traces of an eavesdropped exchange. A typical example is recovering the least significant bit of the static identifier (see for instance [99], the first traceability attack on SASI).

More importantly, an attacker is sometimes able to recover the full static secret after a few rounds. Indeed, she can combine the different observations using Bayesian inference. An example of such an attack was the full cryptanalysis of SASI [10].

4.5.2 Key Updating

One of the initial goals of synchronized protocols is to provide forward privacy. Forward privacy is a stronger notion than privacy. Simply put, a protocol is said to be forward private if an attacker, having recovered the internal state of a tag, is not able to recognize the tag in past interaction traces. For a more formal definition, see [89]. Forward privacy cannot be achieved in a protocol if the secrets used in the exchange are all static. Indeed, if the attacker knows the secrets of a tag at some point, it also knows them in the past, since the secret does not change in the tag’s lifetime. Therefore, she can recompute the messages sent by a tag in previous interactions, and recognize it easily. Note that a changing secret is required for forward privacy, but it does not guarantee it (indeed, there are many synchronized protocols that are not private, and therefore not forward private).

A positive side effect of changing the secrets is that it is conceivably harder to obtain the full secret at any given time, if only a partial leakage is obtained at every authentication round. This seems to be a good feature as it is intuitively harder to hit a moving target that a static one. However, this does not make the full cryptanalysis impossible, just slightly harder, as has been demonstrated with the Tango attacks described in Section 4.7.3.

4.6 Desynchronization

Desynchronization attacks are active attacks. We study them here despite some ultralightweight protocols not claiming resistance against active attackers, only against passive ones. Nevertheless, as we point in Section 3, active attacks are particularly relevant in the considered context. In any case, most authors aim to achieve resilience against active attacks, including some defense against desynchronization.

Desynchronization generally occurs when and active attacker is able to stop the prover and/or the verifier to engage in further successful executions of the protocol. It is generally achieved by tricking the prover, the verifier, or both, into believing that a successful authentication session has taken place, thus updating internal counters and variables in an asynchronous way. Common implementations of this attack involve sending specially crafted messages to make only one of the involved parties reach an irreversible state from which it will not be capable of communicating in the future with other parties.

Many ultralightweight authentication protocols have been shown to be vulnerable to attacks of this kind. A widespread defense mechanism consists in storing the last state (pseudonym and keys) used, together with the current one, and use the former if the latter fails (as first suggested in LMAP [93], by using a flag that is activated when an authentication session terminates abnormally). This approach however modifies the behavior of the protocol, which opens the door to other attacks such as timing attacks [11]. Moreover, it comes with the cost of storing an extra copy of the state, which is relatively expensive in terms of area. Although not exactly flawed, this solution is not perfect.

Desynchronization protection in ultralightweight authentication protocols is relatively poorly studied on its own, and in any case caution is recommended when analyzing the security against such attacks.

An example of two classical desynchronization attacks can be found in [109]. Desynchronization can be considered as a type of Denial of Service attack, and as most DoS attacks, it does not admit an easy, ideal solution.

4.7 Vulnerability to Generalist Attacks

There are only a handful of generalist attacks published in the relevant literature that have proven to be successful against ultralightweight protocols. By generalist we refer to attacks that are not based on ad-hoc methods, and that are then of application against a large class of ultralightweight protocols. We proceed to describe them in some detail on the following sections.
4.7.1 Recursive Linear and Differential Cryptanalysis

In 2013, Ahmadian et al. introduced in [1] a new framework for implementing full disclosure attacks against ultralightweight protocols based on the new concepts of recursive linear and recursive differential cryptanalysis. Both exploit the use of triangular functions, and recover all secrets in a recursive manner. The linear version of the recursive attack is applied to the Yeh-Lo-Winata and SLMAP protocols after passively eavesdropping just one authentication session. On the other hand, the differential version of their recursive attack is deemed to be even more powerful and can be successful against protocols immune to the linear attack, for example LMAP++ and SASI. However, the differential attack is active as the attacker needs to block some exchanged messages, and also requires more (around 13) consecutive authentication sessions.

4.7.2 Gröbner basis attacks

Daewan Han published in 2011 [53] new attacks against lightweight RFID authentication protocols that employed Gröbner basis. He argued his attacks were superior to previous ones because, by not using any specific characteristics of target protocols, they are generally applicable to a variety of them. In addition, he showed his approach was able to recover most secret bits of protocols such as LMAP, M2AP, EMAP, SASI, Lo et al.’s protocol, and Lee et al.’s protocol. The computations needed only a few seconds on a single, standard PC. His approach needed anything between 1 and 5 eavesdropped sessions, took between 0.4 and 151.8 seconds, and recovered from 94.9 to 100% of the secret bits. The author carried out his attacks with the help of the software package PolyBori, implemented in Sage. The major contribution of this paper is that the author presented overwhelming evidence that Gröbner basis are a very powerful toolkit for the cryptanalysis of ultralightweight protocols. The attacks presented in the previous two sections were published much later that the following two attacks, and were no doubt influenced by them, so we will describe these two earlier attacks in some more detail in the following.

4.7.3 Tango Attack

The Tango Attack was first introduced in [57] and [88], and later employed in [19]. Its concept is extremely simple, but it is notable for being another generalist attack. It could also be seen as a new tool to analyze lightweight protocols, and thus helpful in the design of more secure future proposals. The Tango attack is successful against other lightweight protocols, apart from those it was shown to break in the literature. This is a direct consequence of designers not having strict constraints on the resources needed for an adequate (i.e. highly nonlinear) mixture of the internal secret values in the message composition. This makes it hard to avoid leaking some secret bits in every session. The Tango is an attack that aims to obtain a full disclosure of all secrets that the protocol is designed to conceal. It is a passive attack that generally only needs to eavesdrop a small number of (possibly consecutive) authentication sessions, as shown in Figure 4. This is a very realistic attack scenario. The Tango attack is a simple, yet powerful technique of cryptanalysis which is based on the computation and full exploitation of multiple approximations to said secret values, using Hamming distances and the representation of variables in an n-dimensional space. Its first use was against the David and Prasad protocol [30], and it emerged as a passive (i.e. completely realistic in the underlying security model) and extremely efficient attack to fully recover the secret key values $K_1$ and $K_2$ and the static identifier of the tag $ID$.

The attack is divided into two main phases: 1) Selection of good approximations; and 2) Combination of these good approximations for disclosing $K_i$ or $ID$. We briefly describe the two phases below.

Phase 1: The attack exploits the leakage of secret information over the insecure radio channel due to fact that exchanged messages are derived from secret values by using relatively simple (i.e. not highly nonlinear) operations only. This is why the attacker can find and succeed in using multiple simple combinations of the exchanged public messages $f(A, B, D, E, F)$ as good approximations for the secrets $K_i$ or $ID$. In weak protocols, public exchanged messages do not hide these secret values well enough. From all the set of approximations, the adversary is interested on those that are systematically closer (on average) to the target secret value. That is, those for which the Hamming distance between an approximation $Z$ and the value $X$ deviates from the expected value 48, so either $H(Z, X) < 48$ or $H(Z, X) > 48$ significantly.

Phase 2: The basic idea at this stage of the attack is to combine multiple approximations obtained in different sessions, to construct a global one which is highly correlated with the secret values (i.e. keys $K_i$ and static identifier $ID$). This can be done in a number of different ways and forms, but for instance in the case of the David-Prasad protocol, even a simplistic approach works quite nicely. The procedure is the following: for each authentication session eavesdropped, a number of good approximations to the secret values is computed, and then stored as rows of three different matrices. After eavesdropping a given number of sessions, the global values are computed simply by repeatedly adding each of the columns of the matrices, and returning a 0 if the total number of ones in the said column is below a given threshold, or a 1 otherwise. The simplest way to obtain a final value is to select the majority value in each column of this matrix. One can quickly sum all the rows to obtain a final vector. Then, if the value in a column of this vector is greater than half of the number of approximations times the number of eavesdropped sessions, a 1 is conjectured in that column, or a 0 otherwise.

In the case of the David-Prasad protocol, this easy and efficient way of combining approximations works surprisingly well for producing accurate global approximations to all three secret values after eavesdropping a relatively small number of authentication sessions. Results are powerful enough to consider the protocol completely broken, as after observing only 5 or 10 sessions, an attacker can guess about 80 out of 96 bits of the keys and static identifier, as shown in Figure 4. The remaining ones can be easily identified by an offline brute force search, or by eavesdropping more sessions. These results contradict the security properties claimed in [30].

After conducting the attack, the adversary is able to retrieve all the secret information shared between the tag and the server, so she can trivially bypass any authentication mechanisms (i.e. tag and reader authentication) and impersonate the tag in the future, or just clone it. Confidential information is put at risk and tag’s answers can be tracked.
even though two random numbers are used in each session. A desynchronization attack against the tag (or the sever) is also quite straightforward, since the adversary can generate any desired valid synchronization messages.

In [19], a technique to automatically find new good approximations based on the use of genetic programming is described. This further automates the whole Tango attack, and makes it suitable for testing new proposals in a generalist manner.

The Tango attack could be regarded as a close version of the linear cryptanalysis attack by Matsui [55], in which the attacker tries to find a linear approximation for the cipher with maximum possible bias, and extract secret information once enough data is available to her. The main difference between the Tango and linear cryptanalysis is that the former is not limited to linear combinations of its inputs and is, in this sense and in this sense alone, more general than the latter.

4.7.4 Heuristic Simulation Attacks

The concept behind heuristic simulation attacks, again one of the few general-purpose published attacks against ultra-lightweight protocols, is very simple: To infer all or part of the secrets used in a protocol by extracting information from the proximity between the exchanged messages in a real protocol run and those generated by an approximation to the secrets.

These attacks generally start from a randomly generated set of secret values. Then, by applying some heuristic techniques (like simulated annealing) it tries to approximate the true value of the secrets. For that it employs a fitness function that is generally a function of the distance between generated messages and the eavesdropped ones.

Different degrees of success in this approximation could lead to a key recovery attack (none published at the moment, but likely to be possible against weaker protocols) or to a traceability attack (when only some parts of the secrets can be recovered). Of course, a number of different heuristic techniques can be used ranging from hill climbing or simulated annealing to genetic algorithms. The fitness function is generally a measure of the distance between two sets of messages, but can be quite more complex, like weighing some messages more than others, etc.

The first application of this approach was in [58], [59], which presented a traceability attack against SLMA [76]. Authors showed a new meta-heuristic-based traceability attack on SLMA and analyzed its implications. The main interest of their approach is that it is a complete generalist technique that does not make any assumptions on the components of the underlying protocol and can thus be easily generalized to analyze many other proposals.

The fitness function used is shown in Equation (3).

\[
f_S = \sum_{i=0}^{N-1} H(M_i \oplus M'_i) \cdot L^N,\quad (3)
\]

where \(H(\cdot)\) is the Hamming weight, \(M'_i\) are approximations to the real public messages \(M_i\), \(L\) is the length of each message, and \(N\) is the number of observable messages.

The main idea behind this approach is to transform the cryptanalysis of a security protocol into a search problem, where a large number of different search meta-heuristics can be applied. In general, during this search one tries to find which are the secret state values (keys, nonces, etc.) of some subset of the parties involved in the protocol.

This could be done in various ways, but the most natural approach is to measure the cost of the tentative set of secret values by the proximity of the messages produced by these tentative solutions to the real public messages generated and exchanged during the actual protocol execution, as in Equation (3).

Most cryptographic protocols should exchange one or more messages to accomplish their intended objective(s) (authentication, key exchange, key agreement, etc.), and in the vast majority of cases these messages are sent via an insecure or public channel that can be easily snooped.

In this attack model, the cryptanalyst generally tries to infer the secret values that the two parties intend to hide by exploiting the knowledge of the exchanged messages. In a robust, secure, well-designed cryptographic protocol, even states that are very close to the real state should not produce messages that are very close (for any useful distance definition) of the real public messages.

This should be done, typically, by means of a careful design and message construction based on the use of some highly-nonlinear cryptographic primitives such as block ciphers or hash functions.

Unfortunately, new proposals in the field of lightweight cryptography, which are intended towards very computationally constrained environments (such as low-cost RFID systems) cannot use classical cryptographic primitives.

Then, for this search problem, a number of meta-heuristic techniques can be used to minimize the distance between the candidate and the real exchanged messages. In [59], the authors used a Simulated Annealing technique, which is a meta-heuristic technique that is extremely efficient and has some ability to avoid becoming quickly trapped in local minima.

With this, the next IDS can be predicted with sufficient accuracy despite not knowing the protocol secrets, and hence to mount a successful traceability attack. One important characteristic of their attack is that it was successful after eavesdropping only one authentication session, which is a very economic requirement compared with those of other passive attacks.

The general traceability attack algorithm is described in Algorithm 1.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2492553, IEEE Transactions on Mobile Computing

4.7.5 Use

These attacks are quite simple, powerful, and attractive due to their generality and potential. We strongly encourage that new proposals be tested against them to avoid major pitfalls before publication.

Resilience against all these attacks will of course not prove that the security of the new protocol is foolproof, but at least will offer guarantees that the scheme has avoided some of the trivial security weaknesses that plague most of current proposals.

5 Dubious Proofs of Security

We to examine in the following some of the many security proofs that different authors have used over the years in an attempt to prove the security of their proposals.

5.1 Randomness Tests

As shown in Figure 2, one of the first proposals in this area, LMAP, tried to prove some degree of security by verifying that the exchanged messages looked random enough. For that, multiple sessions of the protocol were run and the exchanged messages recorded and later analyzed with different randomness test batteries such as the well-known ENT [118], Diehard [84] and NIST [107].

The results were indeed impressive, with the messages used for running the protocol passing all tests with flying colors. However, this does not prove any security level, as LMAP was broken shortly afterwards. This simple methodological approach presents a number of limitations. Let us have a deeper look on the protocol, as shown in Figure 2.

First, the most obvious one is that we are injecting into the messages two fresh random numbers, \( n_1 \) and \( n_2 \), generated by the reader, and this makes it hardly surprising that some of the exchanged messages, that are combined with these two random messages, look random. Randomness might appear, then, not as a consequence of a well designed protocol but just as a result of employing these sources of randomness repeatedly during message composition. Let us examine an artificial, very simple, and Trivially Weak authentication protocol, depicted in Figure 5.

It is clear that, in this case, an attacker can trivially recover the values of all secrets involved simply by solving the trivial equations involved. Indeed,

\[
\begin{align*}
A &= K_1 \oplus n_1 \\
B &= K_2 \oplus n_2 \\
C &= K_3 \oplus n_1 \oplus n_2 \\
D &= K_1 \oplus n_1 \oplus n_2 \\
E &= K_2 \oplus n_1 \oplus n_2 \\
F &= ID_{tag} \oplus n_1 \oplus n_2 \oplus K_3
\end{align*}
\]

Nevertheless, even in this very simple and weak protocol, exchanged messages pass all randomness statistical tests. At the very least, to prevent the indirect measure of the randomness of the source of \( n_1 \) and \( n_2 \) one has also to examine in detail all the correlations between these exchanged messages.

So if by now it seems clear that randomness of the exchanged messages is not a sufficient condition, it should also be clear that it is neither a necessary one. Another trivial way of showing this is by thinking about highly formatted messages and how, even if a protocol is secure, due to formatting and padding of some or all of its messages these may not pass some randomness test.

So we just have intuitively shown that randomness of the exchanged messages is neither a sufficient nor a necessary condition for protocol robustness, and as such it should no longer be used to prove any kind of security level in future proposals.

4. Provided, of course, that the (P)RNG used for creating \( n_1 \) and \( n_2 \) is good enough.
5.2 BAN and GNY Logic

Some authors have tried to use BAN logic to prove the security of their proposals. A notable example is [102]. Needless to say, this approach did not work out as intended and, despite being accompanied by a formal security proof in BAN logic the proposal was broken shortly afterwards in [92]. Let us analyze this example in more detail: The authors mistakenly employed CRC (Cyclic Redundancy Codes) as recommended by the EPC-C1-G2 standard, but instead of using them as simple error detection tool, they employed them for encryption. In their idealized model, they identified their CRC usage as equivalent to encryption, so some of the BAN logic rules (for example RI: Message-meaning rule) do not hold any more. This is a particularly flagrant mistake because CRCs are linear and should never be used for encryption, hashing or any other cryptographic purposes. But it is also a very common error, as never an idealized scenario like the one modeled by BAN logic (with perfect, unbreakable and zero-leaking ciphers) accurately models reality. The level of abstraction needed in the modeling phase basically makes it impractical for most realistic situations. This is, unfortunately, not only a limitation of BAN logic but, to different extents, is also in most formal models (GNY, etc.) which makes them not very relevant for our purposes.

A recent example of the current and continuous use of these quite limited approaches is [106], where again BAN logic was used again to prove the security of the proposal, despite its severe limitations and unrealistic assumptions. Another recent case can be found in Section 6.3, where a protocol published in Nov. 2013 uses GNY logic to prove its security. Needless to say, this proposal can be attacked quite straightforwardly as mentioned in 5.3.

5.3 Other approaches

Recently, some authors are taking a different approach to this problem, by using automatic analysis tools previously employed on the formal analysis of classical protocols. Two interesting examples are [62] and [112].

In the first, the author uses the AVISPA [3] tool to analyze the security of the LMAP protocol. We have presented multiple vulnerabilities of this protocol along the article, so we are a little wary of the conclusion of this work, where apparently only two attacks have been discovered by AVISPA and the author proposed an easy patch.

Although these results have to be examined with more caution, we believe the general approach is promising and that better proposals will appear if using the AVISPA and similar tools becomes commonplace in the area.

In [112], the authors propose a key establishment and derivation protocol for EPC Gen2 tags, and employ model checking techniques to verify whether the security properties needed hold in a finite state machine. For that, they use automated reasoning, specifically the Constraint-Logic based Attack Searcher (CL-AISec), and the HLPSL input language.

We strongly believe this should be an approach to encourage in the future, and that all new proposals should have been tested by automated tools prior to acceptance. Of course attacks could still exist after this check, but at least we will avoid proposals that fall flat on well-known mistakes from the past.

6 Weaknesses in Recent Protocols

In this section, we present three recently published ultra-lightweight protocols, and propose attacks on them. These attacks are new as far as we know, and illustrate the discussions of Sections 4 and 5 by highlighting weaknesses in their design.

6.1 Bilal and Martin

Bilal and Martin proposed in September 2013 a protocol called UMAP [21]. For reference, the protocol is depicted in Figure 5 in Appendix A. It uses modular rotations, so a natural first attempt at analyzing its security is to look at what happens when they behave like the identity. This happens with a non-negligible probability of $L^{-2}$. The relevant equations of the protocol become:

$$A = n_2 + IDS + K + ID + n_1$$
$$B = n_1 + IDS + K + ID + n_2$$
$$IDS_{next} = n_1 + IDS + n_2$$
$$K_{next} = n_2 + K + n_1$$

Note that in particular, we have that $A = B$. We thus have that, with probability $L^{-2}$, an attacker can authenticate in place of a tag, just by knowing the $IDS$ of some target tag (this may be obtained beforehand simply by sending hello to the tag), and by replicating $A$ as sent by the reader. This attack vector is addressed in Sect. 5.3.

Another point regards the function $f$. It is described in [21] as a “lightweight pseudo random function”. If such a function is readily available on the tag, and if it has good security properties, then why bother with the other constructions? As mentioned in Sect. 5 a very basic challenge-

---

Table 2

<table>
<thead>
<tr>
<th>Statistic</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy (bits/byte)</td>
<td>7.999999</td>
<td>7.999999</td>
<td>7.999999</td>
<td>7.999999</td>
</tr>
<tr>
<td>Compression Rate</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>$\chi^2$ Statistic</td>
<td>250.98 (50%)</td>
<td>255.71 (50%)</td>
<td>244.46 (50%)</td>
<td>255.18 (50%)</td>
</tr>
<tr>
<td>Arithmetic Mean</td>
<td>127.5082</td>
<td>127.4977</td>
<td>127.4946</td>
<td>127.5030</td>
</tr>
<tr>
<td>Monte Carlo $\pi$ Estimation</td>
<td>3.1413 (0.01%)</td>
<td>3.1413 (0.01%)</td>
<td>3.1417 (0.01%)</td>
<td>3.1413 (0.01%)</td>
</tr>
<tr>
<td>Serial Correlation Coefficient</td>
<td>-0.000040</td>
<td>-0.000010</td>
<td>-0.000077</td>
<td>-0.000036</td>
</tr>
<tr>
<td>Diehard Battery (Overall $p$-value)</td>
<td>0.227765</td>
<td>0.775516</td>
<td>0.641906</td>
<td>0.410066</td>
</tr>
<tr>
<td>NIST Battery</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
response protocol, such as the following, works just as well:

\[
\begin{align*}
R & \rightarrow T : \text{hello} \\
T & \rightarrow R : \text{IDS} \\
R & \rightarrow T : r, f_K(r) \\
T & \rightarrow R : f'_K(r)
\end{align*}
\]

6.2 Jeon and Yoon

Jeon and Yoon presented in October 2013 their protocol \textsc{RAPLT} \cite{63}. This protocol (shown in Figure 7 in Appendix A for reference) introduces two new lightweight operations: \textsc{Mer}(A, B, K, C) and \textsc{Sep}(C, K, A, B). Their definition is the following: \textsc{Mer}(A, B, K, C) merges the values \(A \in \{0, 1\}^L\) and \(B \in \{0, 1\}^L\) into \(C \in \{0, 1\}^{2L}\) using the bits of \(K \in \{0, 1\}^L\): if a bit in \(K\) is 0, then a bit of \(A\) is moved to \(C\), else a bit of \(B\) is used. \textsc{Sep}(C, K, A, B) does the opposite: if a bit in \(K\) is 0, the next bit in \(C\) is moved to \(A\), else it is moved to \(B\).

The first thing to note is that these operations are ill-defined as they both require that \(H(K) = L\). The authors make no mention of any way to guarantee that, and it would anyway result in a smaller entropy. Regardless, we assume that this property is satisfied in what follows.

The usage of \textsc{Sep} in the protocol does not follow the definition. The protocol specifies \textsc{Sep}(\textit{M1}, \textit{M2}, \textit{K2}||\textit{K1}, B1||B2), but we assume that the authors meant \textsc{Sep}(\textit{M1}||\textit{M2}, \textit{K2}||\textit{K1}, B1, B2), since else the variables would not have the right sizes.

Another point is that these operations, that consist in rearrangements of bits of some of their inputs, have linear properties. The only other operation used is the \textsc{XOR}, which makes the protocol entirely linear. This problem is addressed in Sect. 4.1.

Finally, a traceability attack may easily be mounted against the protocol by using the fact that the new operations are Hamming-weight invariant. A similar attack was previously done on \textsc{RAPP} \cite{111} in \cite{9}. We have that, after \textsc{Mer}(A, B, K, C), or after \textsc{Sep}(C, K, A, B),

\[
H(A) + H(B) = H(C) = H(C_1) + H(C_2),
\]

and we also have that

\[
H(A \oplus B) \equiv H(A) + H(B) \pmod{2},
\]

for any \(A, B\). Therefore, we can deduce the following:

\[
\begin{align*}
H(B_3) & \equiv H(B_1) + H(B_2) \\
& \equiv H(M_1) + H(M_2) \\
& \equiv H(N_1) + H(N_2) + H(K_2) + H(K_1) \pmod{2} \\
H(A_1) + H(A_2) & \equiv H(N_1) + H(N_2) \pmod{2}
\end{align*}
\]

Therefore,

\[
H(K) \pmod{2} = (H(B_3) + H(A_1) + H(A_2)) \pmod{2}
\]

Since \(A_1, A_2 \) and \(B_3\) are public, and since \(K\) is static, it is easy for an attacker to trace a tag by observing its communications with a genuine reader.

6.3 Ling and Shen

In an article of November 2013, Ling and Shen \cite{81} present a protocol that is strongly inspired on the SASI protocol \cite{27}. A formal proof using GNY logic is made, which should indicate that the protocol is secure.

However, since messages \(C\) and \(D\) are exactly the same ones as in SASI, Phan’s simple traceability attack \cite{99}, which allows a passive attacker to recover one bit of the secret, works here too. Without going into the details, it essentially relies on the fact that the OR (\(\lor\)) operation has biased output, an issue that has been covered in Section 4.2.

This attack proves that the formal security proof made in \cite{81} is not valid, a point that was already raised in Section 5.2.

7 Promising Trends

7.1 Ultralightweight Primitives

Rather than designing an authentication protocol from scratch using ultralightweight operations, another (arguably more reliable) approach to designing an ultralightweight authentication protocol is to use a classical challenge-response authentication protocol, and employ an ultralightweight construction as the underlying cryptographic primitive.

In the category of ultralightweight primitives, that goes up to around 1.5K GE, there is generally no room for highly optimized versions of standard ciphers like AES and IDEA, or slight modifications of classical block ciphers like DES (DES\textsc{L}, DES\textsc{XL}) \cite{70}.

Only newly and purposefully designed low-cost primitives can generally be ascribed to it. Fortunately for the area, more and more of the new proposals fall within this limit. There seems to be a growing consensus that this is about the maximum GE that most devices would be able to devote to security.

Within this upper limit, there was not any proposal at all before the publication of \textsc{PRESENT} \cite{23} in 2007, as both DES (2300-3000 GE) or even DES\textsc{XL} (2168 GE) are too large. AES needs even more gates, with around 3600 GE (recent results \cite{38} put this at 2400GE, still well above the 1.5K limit). Not even well-known extremely efficient and even easy to memorize algorithms such as \textsc{TEA} \cite{119} (2100) and \textsc{XTEA} (2000) were light enough. Other recent proposals like \textsc{MCRYPT} \cite{29} (2949 GE), \textsc{HIGHT} \cite{60} (3000) and \textsc{SEA} \cite{108} (2280) were also not adequate for our purposes.

After 2007, on the other hand, we have witnessed a blossoming of new primitives. This new trend includes most notably \textsc{PRESENT} (1570 GE) that, based on the AES finalist \textsc{Serpent}, can rightly be considered a pioneer in this area.

The European eSTREAM project was also relevant, as it looked for stream ciphers suitable for efficient hardware implementation, and it contributed to the development of \textsc{Trivium} (2599 GE) and \textsc{Grain} (1294 GE). Of course only \textsc{Grain} \cite{54} is adequate for our intended environments.

Continuing with ultralightweight block cipher proposals, we have \textsc{LBlock} (2011) which uses 1320 GE, but also many other recent algorithms like the block cipher \textsc{TWINE} (1116 GE), \textsc{MiBS} (1400), \textsc{KATAN}/\textsc{KTANTAN} \cite{31} (from 688), \textsc{Piccolo} (683 to 1043), \textsc{LED} \cite{47} (1040) and the \textsc{PRINCIples} (503) or \textsc{PRINCE}.

From this long list, specially notable are the \textsc{KATAN} and \textsc{LED} proposals, which have gained some respect and popularity after resisting cryptanalysts’ efforts for a time.
Although all have received some minor attacks, these look at the moment like a particularly good trade-off between security and efficiency.

It is important to note that we have to be cautious on any comparison based exclusively on gate count, because not all of these algorithms operate over the same keysize, blocksize or produce the same throughput.

Lastly, it is significant to note that even the NSA came with two proposals in this area, called SIMON (763 GE) and SPECK (884 GE). These were meant to provide high performance across a range of devices. In the light of recent NSA scandals these two proposals, independently of their respective merits, will probably never prove very popular within the cryptography community. On top of that, there are some attacks that present a not particularly rosy view on either of them.

But not only block and stream ciphers designers have been attracted to the challenge of devising ultralightweight primitives. There are also some interesting, though less numerous, proposals in other areas of cryptography. Particularly attractive are the SPONGENT [23] (from 738 GE to 1950) and QUARK [7] (from 1379 to 4460 GE) families of lightweight hash functions. These two, despite being by far the most popular, are not the only ones available as there is also the PHOTON [46] family (1122 to 2768 GE).

We believe this is one of the most promising trends in ultralightweight cryptography, and one of the more sound approaches to provide in the near future with adequate security solutions for low-cost devices. We encourage more work on the area, both in proposing new schemes and in assessing the security of existing ones. For speeding up this process, we believe that a new AES or SHA-3 like competition would be very helpful to set new ultralightweight standards primitives. It would probably be very well received, and would no doubt contribute greatly to advance the state of the art and to progress in the widespread adoption of these or new proposals.

7.2 Human-Based Protocols

Juels and Weis introduced in 2005 an authentication protocol, known as HB+ [64], which is based on HB, a secure identification scheme for Human beings designed by Hopper and Blum in 2001 [61]. HB+ requires only lightweight calculations from the prover and the protocol benefits from a security proof based on a reduction to the Learning Parity with Noise (LPN) problem: let $x \oplus b$ a secret k-bit vector, $a$ a random known k-bit vector, $\epsilon \in \mathbb{F}_2$ a noise parameter, and $\eta$ a bit noise where $\Pr(\eta = 1) = \epsilon$; it is hard to learn $x$ from the value $r = a \cdot x \oplus \eta$.

In spite of attractive properties, analyses of HB-based protocols should be done with the utmost care. Indeed, while HB+ benefits from security proofs in restricted security models [64], [66], the protocol becomes insecure when considering active adversaries who can modify the messages exchanged between the involved parties. Gilbert, Robshaw, and Sibert pointed out this weakness in [44] (2005). Many attempts to make HB+ secure against man-in-the-middle attacks were published, but Gilbert, Robshaw, and Seurin then raised flaws in 2008 [43] in the published variants of HB+ namely HB++ [26] (2006) and a variant by Piramuthu [101] (2006), HB’ [64] (2007), HB-MP and HB-MP+ [87] (2007).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2492553, IEEE Transactions on Mobile Computing 11


In parallel, improvements on HB-MP were suggested: HB-MP+ [73] (2008), HB-MP++ [122] (2009), and HB-MP" [6] (2011), and more recently HB-MP received improvements from Tsaur et al. in [60] (2011) and [113] (2012).

Other variants include Trusted-HB [25] (2008), demonstrated to be insecure by Frumkin and Shamir in [40] (2009), NLHB and NLHB+ [83] (2010), a bilinear version of HB named HBN [24] (2011), and a construction by Kiltz et al. [68] (2011) based on Pietrzak’s work [100] that is still based on the LPN problem but which significantly differs from the HB family. Finally, Rizomiliotis and Gritzalis introduced GHB+ [105] (2012) to make secrets shorter than the ones needed in [68].


As a consequence, any HB-related authentication scheme should be analyzed considering the previous unsuccessful attempts. Two references should also be the keystones of the analyses, namely [43] and [90].

Finally, note that HB and its variants are considered not so lightweight by some experts, and in particular do not fit within the 1500 GE margin. See [5] for a thorough discussion on the subject.

8 Conclusion

In this article we described the state of the art of ultralightweight protocols security, and tried to address some of its many issues. We showed that virtually all protocols are broken, with half of them having been broken within 8 months and most of them within a year of their publication. We reviewed the most critical and most frequent mistakes encountered in these protocols and illustrated them on three very recent proposals. These new attacks show that little effort is often sufficient to break such schemes, and that the topics addressed in this work are relevant for the analysis of novel proposals.

Our work provides guidelines to quickly assess the security of new proposals and should additionally help reviewers to efficiently distinguish those of merit. Likewise, authors of new ultralightweight protocols could review the points addressed in this article in order to evaluate the correctness of their claims.

Despite the current situation, which may seem to indicate that ultralightweight protocols are bound to fail, there is no clear evidence that designing a secure protocol with such constraints is impossible. Much to the contrary, recent advances in lightweight cryptography present an optimistic future for research in the area. In particular, we have highlighted in Sections 4.7.1, 4.7.2, [7] the current developments that, in our opinion, show greater promise.

We are hopeful that our work will help to improve the quality of future proposals in the field, avoiding common
past and present mistakes, and help the area mature towards the creation of new standards.

REFERENCES


[14] Nasour Bagheri, Praveen Gauravaram, Masoumeh Safkhani, and Somitra Kumar Saradnya. The resistance to intermittent position trace attacks and desynchronization attacks (RIPTA-DA) protocol is not RIPTA-DA. In Workshop on RFID Security – RFIDSec’13, volume 8262 of Lecture Notes in Computer Science, Graz, Austria, July 2013.


[41] Lijun Gao, Maode Ma, Yantai Shu, and Yuhua Wei. A security protocol resistant to intermittent position trace attacks and desynchronization attacks in RFID systems. Wireless Personal Communications, pages 1–17, July 2012.


Gildas Avoine is a professor of information security and cryptography at the INSA Rennes (France). He is also an invited professor at the UCL in Louvain-la-Neuve (Belgium) and a member of the Institut Universitaire de France. Previously, he was a researcher at the MIT (USA) and at the EPFL (Switzerland), where he obtained a PhD degree in cryptography. He also studied at the University of Gaen (France) where he received a Bachelor degree in mathematics and Bachelor and Master degrees in computer science.

Xavier Carpent obtained his PhD from the UCL in Louvain-la-Neuve (Belgium) in 2015. He graduated as an engineer in computer science at the UCL and the UPO (Barcelona). His primary research interests include RFID authentication protocols and time-memory trade-offs.

Julio Hernandez-Castro is a Lecturer in Computer Security at the School of Computing, University of Kent, Canterbury Campus, in the UK. He obtained his PhD in 2003 in Computer Science from Carlos III University in Madrid, Spain. His main research interests are in RFID authentication protocols, Lightweight Cryptography, CAPTCHAs, Steganography, Steganalysis and Network and Computer Forensics.
A

APPENDIX

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46</td>
<td>4</td>
<td>60</td>
<td>0.23591</td>
<td>-0.00532</td>
<td>Success</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>13</td>
<td>60</td>
<td>0.24760</td>
<td>-0.08020</td>
<td>Success</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>14</td>
<td>33</td>
<td>0.10418</td>
<td>0.02094</td>
<td>Success</td>
</tr>
<tr>
<td>4</td>
<td>39</td>
<td>11</td>
<td>36</td>
<td>0.13681</td>
<td>-0.01610</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>44</td>
<td>6</td>
<td>59</td>
<td>0.22318</td>
<td>-0.03496</td>
<td>Success</td>
</tr>
<tr>
<td>6</td>
<td>34</td>
<td>16</td>
<td>55</td>
<td>0.10919</td>
<td>-0.12903</td>
<td>Success</td>
</tr>
<tr>
<td>7</td>
<td>41</td>
<td>9</td>
<td>56</td>
<td>0.17157</td>
<td>0.12330</td>
<td>Success</td>
</tr>
<tr>
<td>8</td>
<td>43</td>
<td>7</td>
<td>62</td>
<td>0.29247</td>
<td>0.08456</td>
<td>Success</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>5</td>
<td>60</td>
<td>0.24967</td>
<td>-0.06447</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>42</td>
<td>8</td>
<td>58</td>
<td>0.19593</td>
<td>0.19375</td>
<td>Success</td>
</tr>
<tr>
<td>11</td>
<td>44</td>
<td>6</td>
<td>58</td>
<td>0.21009</td>
<td>0.09940</td>
<td>Success</td>
</tr>
<tr>
<td>12</td>
<td>35</td>
<td>15</td>
<td>53</td>
<td>0.10707</td>
<td>-0.11736</td>
<td>Success</td>
</tr>
<tr>
<td>13</td>
<td>38</td>
<td>12</td>
<td>53</td>
<td>0.09923</td>
<td>-0.02172</td>
<td>Success</td>
</tr>
<tr>
<td>14</td>
<td>47</td>
<td>3</td>
<td>61</td>
<td>0.036238</td>
<td>-0.04929</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>35</td>
<td>15</td>
<td>53</td>
<td>0.10629</td>
<td>0.14811</td>
<td>Fail</td>
</tr>
<tr>
<td>16</td>
<td>47</td>
<td>3</td>
<td>66</td>
<td>0.36751</td>
<td>-0.16724</td>
<td>Success</td>
</tr>
<tr>
<td>17</td>
<td>44</td>
<td>6</td>
<td>59</td>
<td>0.21093</td>
<td>0.03115</td>
<td>Success</td>
</tr>
<tr>
<td>18</td>
<td>39</td>
<td>11</td>
<td>55</td>
<td>0.14885</td>
<td>0.03840</td>
<td>Success</td>
</tr>
<tr>
<td>19</td>
<td>31</td>
<td>19</td>
<td>52</td>
<td>0.10013</td>
<td>-0.01543</td>
<td>Success</td>
</tr>
<tr>
<td>20</td>
<td>39</td>
<td>11</td>
<td>58</td>
<td>0.20798</td>
<td>0.03845</td>
<td>Success</td>
</tr>
</tbody>
</table>

Averages 40.3 9.7 57.25 0.19 0.01 95% Success

A

Figure 6. Bilal and Martin’s protocol.
Figure 7. Jeon and Yoon’s protocol.