SLAP: Secure Lightweight Authentication Protocol for Resource-constrained Devices

Giulio Aliberti[1], Roberto Di Pietro[2,3] and Stefano Guarino[4]
[1]*Department of Mathematics and Physics, Roma Tre University, Rome, Italy*
[2]*Cyber Security Research Department, Nokia Bell Labs, Paris, France*
[3]*Department of Mathematics, University of Padova, Padua, Italy*
[4]*Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche, Rome, Italy*

Keywords: Light-weight Authentication, IoT, Probabilistic Security.

Abstract: Motivated by the upcoming Internet of Things, designing light-weight authentication protocols for resource constrained devices is among the main research directions of the last decade. Current solutions in the literature attempt either to improve the computational efficiency of cryptographic authentication schemes, or to build a provably-secure scheme relying on the hardness of a specific mathematical problem. In line with the principles of information-theoretic security, in this paper we present a novel challenge-response protocol, named SLAP, whose authentication tokens only leak limited information about the secret key, while being very efficient to be generated. We do support our proposal with formal combinatorial arguments, further sustained by numeric evaluations, that clarify the impact of system parameters on the security of the protocol, yielding evidence that SLAP allows performing a reasonable number of secure authentication rounds with the same secret key.

1 INTRODUCTION

The advent of the Internet of Things (IoT) (Atzori et al., 2010) is leading to a scenario where a multitude of objects with constrained computational capabilities, such as RFID tags (Vogt, 2002) or wireless sensors (Gubbi et al., 2013), are necessarily equipped with identification systems. Additionally, low-end hardware authentication tokens (Weir et al., 2009) and, more generally, RSA SecurID tokens (RSA, The Security Division of EMC, 2015) are increasingly used to realize two-factor authentication, *e.g.*, for e-banking services. In this context, *light-weight* authentication protocols can be used to protect data from unauthorized and potentially harmful access, while minimizing computational burden, time delay, and energy consumption.

Most authentication protocols rely on a challenge-response mechanism, in which valid responses can be produced based on a secret key $\mathbf{s}$ shared between client and server. A secure scheme must ensure that, if $\mathbf{s}$ is unknown, producing a valid response is infeasible, either computationally of probabilistically. However, the only scheme that guarantees *perfect* secrecy is based on the well-known One-Time Pad (OTP) (Shannon, 1949): an n-bit symmetric secret key $\mathbf{s}$ is split into N sub-keys $\mathbf{s}_1, \ldots, \mathbf{s}_N$ of length n/N, providing N authentication rounds each with security parameter $\varepsilon = 2^{-n/N}$. To mimic the security of OTP-based schemes, RSA SecurID tokens (RSA, The Security Division of EMC, 2015) replace the initial secret key with a short seed and sub-secret keys (*e.g.*, authentication codes) are generated from the seed using AES-based cryptography. RSA SecurID tokens offer strong security properties relying on computational security (*i.e.*, the security of the key derivation mechanism).

Recent research efforts aiming at developing light-weight and practical authentication protocols mostly resorted to extensions and variants of the Hopper-Blum (HB) protocol (Hopper and Blum, 2001). HB is a challenge-response authentication protocol based on linear coding: the n-bit symmetric key is "encoded" through a set of n-bit challenges playing the role of parity equations; the obtained "codeword" is then bitwise xored with some pseudo-random noise to produce the response. To pass the authentication test, at least some fixed fraction (*e.g.*, the majority) of the response bits must be correct, *i.e.*, noiseless. Despite their simplicity, the security of all HB-based schemes is again based on computational security, specifically on the assumption that decoding linear codes is computationally hard in the presence

Aliberti, G., Pietro, R. and Guarino, S.
SLAP: Secure Lightweight Authentication Protocol for Resource-constrained Devices.
DOI: 10.5220/0006431101630174
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 163-174
ISBN: 978-989-758-259-2

of noise, a problem known as *Learning Parity with Noise* (LPN) (Pietrzak, 2012).

In this paper, we introduce SLAP, a novel light-weight solution that combines ideas from the aforementioned classes of protocols: inspired by OTP-based authentication, SLAP trades memory for achieving security, neither involving CPU-heavy computations nor relying on computational limitations of the adversary[1]; similarly to HB, SLAP's system parameters are tuned so as to make legitimate responses discernible from random ones, while concurrently guaranteeing that challenge-response pairs leak very limited information about the key. With respect to OTP-based authentication, SLAP benefits from each key bit having a positive probability of being involved in each authentication attempt, thus offering increased flexibility and resilience to common side channel and physical attacks. As a counterpoint, to obtain the desired level of security SLAP relies on multiple challenges per authentication round, with consequent communication/computation overhead.

Albeit all building blocks of SLAP are extremely efficient, a direct comparison with previous solutions in terms of resource consumption, useful to clarify SLAP's most suitable application setting, would require a detailed analysis of memory/cpu/energy footprints that lies beyond the scope of the present paper and is left to future work. Yet, in this paper we carefully assess the relation binding security and system parameters, to provide clear evidence, both theoretically and experimentally, that SLAP permits a number N of secure authentications, where N depends on system and security parameters. Although our protocol does not guarantee perfect secrecy, because the entropy of the key slowly decreases with the number of intercepted challenge-response pairs, we do present combinatorial arguments in support of the soundness of SLAP. Significantly, numerical evaluations of our analytical findings allow identifying a few concrete use cases in which the performance of SLAP is comparable to OTP-based authentication, in terms of volume of secure authentication rounds per key.

Roadmap: related work is reviewed in Section 2; SLAP is presented in Section 3 and analysed in Section 4; Section 5 reports numerical evaluations that enable tuning all system parameters; finally, Section 6 concludes the paper.

[1]Except obvious ones, such as assuming that a brute force attack is unfeasible.

2 RELATED WORK

In the last few years, the need for light-weight protocols arose from both the growing need to minimize the power consumption of electronic devices (Patton and McGuinness, 2014; Chabarek et al., 2008) and the severe constrains imposed by the upcoming Internet of Things (Gubbi et al., 2013; Raza et al., 2013; Sehgal et al., 2012). Several light-weight cryptographic protocols for resources constrained devices have been proposed (Alippi et al., 2014), aiming at enabling encryption, signature or hashing (Oder et al., 2014; Aumasson et al., 2010; Engels et al., 2010). A few protocols, similarly to our SLAP, are designed explicitly with authentication in mind, and mostly based on a challenge-response mechanism. When symmetric encryption is an option, an extremely fast and communication-efficient solution consists in using block ciphers (*e.g.*, AES (Feldhofer et al., 2005) or HUMMINGBIRD (Engels et al., 2011)) to produce a valid response based on the challenge and the secret key. However, specific scenarios exist where non-cryptographic authentication is required/preferred, either because of resource limitations (*e.g.*, in RFID tags or IoT scenarios), or because authentication is the unique security requirement (*e.g.*, in physical access control (Kao et al., 2011)). A possible non-cryptographic approach, based on the LPN assumption, has its roots in the HB protocol (Hopper and Blum, 2001). Several variants of HB have been proposed, with HB^+ (Juels and Weis, 2005) and $HB^\#$ (Gilbert et al., 2008) being among the most relevant. However, these computationally efficient HB-based protocols have high communication complexity and are, thus, not suitable is many practical applications. More recently, the Lapin protocol (Heyse et al., 2012), relying on a variation of LPN, has been proposed to overcome communication issues but its validity was soon questioned (Gaspar et al., 2014; Bernstein and Lange, 2012). In (Armknecht et al., 2014) it is argued that the design of an LPN-based protocol for low-cost RFID tags is still an open problem. To the best of our knowledge, SLAP is the first attempt to obtain a novel light-weight authentication protocol based on neither the hardness of a mathematical problem nor cryptography.

The main drawback of SLAP is the limited number of authentication rounds allowed with a single key. However, many attacks are currently able to compromise the lifetime of a key: side channels can be used to bypass logical layer protections via, for instance, timing analysis (Kocher, 1996) or differential power analysis (Kocher et al., 1999), and resource constrained low-cost devices may be exposed

to physical attacks, that undermine the security of the key. Key rotation or updating techniques, already applied to resource constrained devices (Bowers et al., 2013) to defend against key compromise, are a possible solution to extend the usability of SLAP, although the actual feasibility of a similar approach depends on the specific application setting. Interestingly, SLAP leverages the whole key for each authentication round, so that partial knowledge of the key, although helping, is not enough to permit a successful authentication.

3 PROTOCOL DESCRIPTION

In this section we present SLAP, an authentication protocol designed for devices with constrained energy and computational capabilities. From a general standpoint, SLAP is a challenge-response protocol whose actors are a client that wants to be authenticated and a server that manages the authentication request.

Notation. Let us first briefly recap the notation used in the reminder of this paper: if $n \in \mathbb{N}$, $[n] = \{1, \ldots, n\}$ is the set of the first n positive integers; the cardinality of a set A is $|A| \in \mathbb{N}$; $\mathbb{F}_2^n$ denotes the set of all n-bit vectors $\mathbf{x} = (x_1, \ldots, x_n)$, $x_i \in \mathbb{F}_2$; if $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, $H_n(\mathbf{x}, \mathbf{y})$ denotes their *Hamming distance*; finally, if $A = \{i_1, \ldots, i_m\} \subset [n]$, π_A denotes the projection of n-bit vectors into the m coordinates identified by A, *i.e.*: $\pi_A : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^m$, $\mathbf{x} \mapsto \pi_A(\mathbf{x}) = (x_{i_1}, \ldots, x_{i_m})$.

Protocol. Now, let us describe SLAP, with the help of its pseudo-code reported in Protocol 1. The parameters of the protocol are four non-negative integers (n, m, k, δ) that satisfy the condition $n \geq m \geq k \geq \delta$. Client and server are assumed to initially agree on a secret key $\mathbf{s} \in \mathbb{F}_2^n$, chosen uniformly at random, and the server has access to a storage device $\mathcal{D}$ used to keep track of already used challenge-response pairs. The protocol starts when a new authentication request is sent by the client to the server (Line 1 and Line 16). The authentication server elaborates the request generating uniformly at random a challenge $\mathbf{c} \in \mathbb{F}_2^n$ (Line 3), satisfying two conditions: (i) $H_n(\mathbf{s}, \mathbf{c}) \in [k + \delta, n - m + k - \delta]$, and (ii) no previous challenge-response pair $(\mathbf{c}_i, R_i)$ exists such that $H_m(\pi_{R_i}(\mathbf{c}), \pi_{R_i}(\mathbf{c}_i)) < k$.[2] Then the server sends the challenge $\mathbf{c}$ to the client (Line 5), asking (Line 6) and waiting (Line 7) for a response. The client, that was waiting for a challenge (Line 17), receives $\mathbf{c}$ and generates a valid response R for it (Line 18). A valid response R must satisfy: (i) $R \subset [n]$, (ii) $|R| = m$, and (iii) $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) \in [k - \delta, k + \delta]$. The client then sends R to the server (Line 19) and waits to receive an acknowledgement (Line 20). When the server receives the response R, based on its validity (Line 8), it accepts (Line 9) or denies (Line 13) the authentication, sending the appropriate acknowledgement to the client (Lines 10 and 14). In the former case, it writes the new pair $(\mathbf{c}, R)$ in $\mathcal{D}$ (Line 11).

[2] This check is *de facto* the only difference between the performance of SLAP at the client and server sides. The storage and computational overhead scale linearly with the number of allowed pairs N.

Protocol 1: SLAP authentication protocol.

Parameters: *non-negative integers* n, m, k, δ *such that* $n \geq m \geq k \geq \delta$.

Initialization: *server and client agree on a secret key* $\mathbf{s} \in \mathbb{F}_2^n$*; the server has access to a storage device* $\mathcal{D}$*, initially empty.*

Server side

1 wait *to receive an* authentication request *from the client*;
2 **do**
3 generate *uniformly at random a* challenge $\mathbf{c} \in \mathcal{F}_{\in}^{\setminus}$;
4 **while** $H_n(\mathbf{s}, \mathbf{c}) \notin [k + \delta, n - m + k - \delta]$ OR $\exists (\mathbf{c}_i, R_i) \in \mathcal{D} : H_m(\pi_{R_i}(\mathbf{c}), \pi_{R_i}(\mathbf{c}_i)) < k$;
5 send $\mathbf{c}$ *to the client*;
6 ask *to the client for a* response $R \subset [n]$ *with* $|R| = m$;
7 wait *to receive the* response R *from the client*;
8 **if** $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) \in [k - \delta, k + \delta]$ **then**
9 authentication succeeded;
10 send *an* authentication success acknowledgement *to the client*;
11 write (c, R) *in the storage device* $\mathcal{D}$;
12 **else**
13 authentication failed;
14 send *an* authentication failure acknowledgement *to the client*;
15 **end**

Client side

16 send *an* authentication request *to the server*;
17 wait *to receive a* challenge $\mathbf{c} \in \mathcal{C}$ *from the server*;
18 generate *uniformly at random a* valid response $R \subset [n]$ *with* $|R| = m$;
19 send R *to the server*;
20 wait *to receive an* authentication acknowledgement *from the server*;

Comments. Defining a lightweight authentication protocol that does not rely on cryptography presents us with two conflictual requirements: to make it difficult to guess R when $\mathbf{s}$ is unknown, and where $\mathbf{s}$, $\mathbf{c}$ and R need to be correlated; yet, if challenge-response pairs can be easily eavesdropped, the conditional entropy of $\mathbf{s}$ given $\mathbf{c}$ and R must be large. To solve this dilemma, we defined a challenge-response mechanism that depends on four parameters, with the underlying idea that m, k and δ can be tailored to the eavesdropping ability of the adversary, while increasing n provides the ultimate defence against brute-force response-guessing attacks.

Informally, a valid response is a set of m indices such that the challenge and the secret key differ on a number $k \pm \delta$ of such indices, and coincide on the others. Eavesdropping on a successful authentication round only opens a *window* on a portion of the key, and the relative size of m with respect to n allows tuning the size of this window to deal with different adversary models. m defines both the amount of information leaked by overheard traffic, and the amount of information needed to forge a response, and it can be neither too small nor too close to n (*e.g.*, $m = 1$ makes random responses valid with probability $1/2$, while $m = n$ forces $R = [n]$). In any case, snooping through the window opened by a response R only gives a blurred view of the corresponding portion of the key, with k and δ determining how blurred. k and δ indeed define mean and variance of $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c}))$ for a valid R, thus k should be kept close to $m/2$ and δ should be large to limit information leakage. However, guessing a valid R is easier if $k \approx m/2$ (for a randomly chosen R, $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) \approx m/2$) and if δ is large (*e.g.*, $\delta = k = m/2$ means all responses are valid). Luckily, all these apparent contradictions can be solved by increasing n enough to make brute-force response-guessing attacks infeasible, even once the relative size of m, k and δ is fixed to limit eavesdropper adversaries. Details on the impact of the four parameters on the security of SLAP will be provided in Section 4, together with a motivation for condition (ii) for the choice of a valid challenge. Condition (i) is only aimed to make sure that the chosen $\mathbf{c}$ does not condition the admitted range for $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c}))$, as stated in Theorem 1.

Finally, to support the efficiency of SLAP, let us suggest a possible algorithm to generate uniformly at random a valid response: (i) compute $\mathbf{y} = \mathbf{s} \oplus \mathbf{c}$; (ii) apply a random permutation φ to $\mathbf{y}$; (iii) randomly select $j \in [k-\delta, k+\delta]$; (iv) select the first j indices $\{a_1, \ldots, a_j\}$ such that $y_{a_l} = 1$ and the first $m-j$ indices $\{b_1, \ldots, b_{m-j}\}$ such that $y_{b_l} = 0$; (v) define $R = \{\varphi^{-1}(a_1), \ldots, \varphi^{-1}(a_j)\} \cup \{\varphi^{-1}(b_1), \ldots, \varphi^{-1}(b_{m-j})\}$.

4 SECURITY ANALYSIS

In this section, we analyse the security of SLAP considering first an oblivious attacker, who tries to get authenticated without any prior knowledge of the key, and later on an eavesdropper attacker, who tries to take advantage of previously intercepted valid challenge-response pairs to forge a correct response to a newly issued challenge. For the sake of readability, all proofs are deferred to the Appendix.

4.1 Oblivious Adversary Model

We here analyse the security of SLAP in the *oblivious adversary model*, a scenario where the attacker has no information about the secret key whatsoever. A similar attacker cannot do anything better than randomly guessing a valid response for the issued challenge (guessing the key would be even harder). The following results determine how hard a response-guessing attack is, as a function of system parameters. First, we find the number of valid challenges.

Theorem 1. *For any secret key $\mathbf{s} \in \mathbb{F}_2^n$ and for any challenge $\mathbf{c} \in \mathbb{F}_2^n$, if the condition*

$$k + \delta \leq H_n(\mathbf{s}, \mathbf{c}) \leq n - m + k - \delta \tag{1}$$

is satisfied, then for each $j \in [k-\delta, k+\delta]$ the challenge $\mathbf{c}$ admits (at least) a valid response R such that $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) = j$. The cardinality of the set of challenges $\mathcal{C} = \mathcal{C}(n, m, k, \delta)$ satisfying (1) is

$$|\mathcal{C}| = \sum_{i=k+\delta}^{n-m+k-\delta} \binom{n}{i}. \tag{2}$$

Next, we determine the number of valid responses.

Theorem 2. *For any secret key $\mathbf{s} \in \mathbb{F}_2^n$, let $\mathbf{c} \in \mathcal{C}$ be such that $H_n(\mathbf{s}, \mathbf{c}) = i$. The number of valid responses to $\mathbf{c}$ is*

$$R(i) = \sum_{j=k-\delta}^{k+\delta} \binom{i}{j} \binom{n-i}{m-j}. \tag{3}$$

In particular, if $\mathcal{P}$ denotes the set of all valid challenge-response authentication pairs, it holds

$$|\mathcal{P}| = |\mathcal{P}(n, m, k, \delta)| = \sum_{i=k+\delta}^{n-m+k-\delta} R(i) \binom{n}{i}. \tag{4}$$

Now, let *Aut* denote the event of being authenticated when the secret key $\mathbf{s}$ and a challenge $\mathbf{c}$ are chosen as prescribed by Protocol 1, and when the response R is chosen uniformly at random among all subsets $R \subset [n]$ such that $|R| = m$. The following results define the security of SLAP under the oblivious adversary model.

Lemma 1. *In the oblivious adversary model, let $\mathbf{s}^{(0)}$ and $\mathbf{c}^{(0)}$ be such that $H_n(\mathbf{s}^{(0)}, \mathbf{c}^{(0)}) = j$. It holds*

$$P\big(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}, \mathbf{c} = \mathbf{c}^{(0)}\}\big) = \frac{R(j)}{\binom{n}{m}} \tag{5}$$

Theorem 3. *In the oblivious adversary model, the success probability of a response-guessing attack is*

$$P(Aut) = \frac{1}{|\mathcal{C}| \binom{n}{m}} \sum_{j=k+\delta}^{n-m+k-\delta} \binom{n}{j} R(j) \tag{6}$$

The closed expression (6) is analysed in Section 5 for different choices of parameters n, m, k, and δ.

4.2 Eavesdropper Adversary Model

In this section we analyse the security of SLAP in the *eavesdropper adversary model*, *i.e.*, against an attacker able to eavesdrop the communications between client and server to earn information about the key. We aim at estimating the probability ε of guessing a valid response for a newly issued challenge after the attacker eavesdropped on $N > 0$ challenge-response cycles. At a high level, we will proceed as follows:

- we find suitable conditions to ensure that the entropy of the key is large enough to avoid key-guessing attacks (Theorem 4);
- we exhibit sample attack scenarios showing that focusing on the overall entropy of the key (*i.e.*, the key pool size) is not sufficient, and that further attention has to be pointed towards the entropy of each individual key bit;
- we identify suitable and general conditions to prevent the adversary from earning a dangerous amount of information about (portions of) the key, also showing how these conditions can be enforced by a proper tuning of system parameters.

Numerical evaluations of our findings are later discussed in Section 5.

4.2.1 Number of Compatible Keys

To ease exposition, let us denote

$$I_N = \big\{(\mathbf{c}_1, R_1), \ldots, (\mathbf{c}_N, R_N)\big\} \in \mathcal{P}^N$$

a set of N valid challenge-response pairs, and

$$S_N = \big\{\mathbf{v} \in \mathbb{F}_2^n : H_m(\pi_R(\mathbf{v}), \pi_R(\mathbf{c})) = k \pm \delta, \forall (\mathbf{c}, R) \in I_N\big\}$$

the restricted key pool induced by eavesdropping I_N, *i.e.*, the set of all keys that are *compatible* with the N challenge-response pairs in I_N. The size of S_N, denoted by Y_N, is a random variable that depends on the specific choice of I_N. Y_N thus gives a first measure of the security of SLAP against an eavesdropper adversary, by quantifying her uncertainty on the key (*i.e.*, the entropy of the key) after discarding non-compatible candidate keys.

Theorem 4. *In the eavesdropper adversary model, after N valid challenge-response pair has been intercepted, the expected size of the key pool is*

$$\mathbb{E}[Y_N] = 2^n \rho^N, \quad \textit{where } \rho = 2^{-m} \sum_{j=k-\delta}^{k+\delta} \binom{m}{j} \qquad (7)$$

Theorem 4 estimates how many vectors of $\mathbb{F}_2^n$, on average, are compatible with N randomly selected authentication pairs. The more N increases, the lower is the number of these vectors. We observe that, for any fixed N, $m \not\sim n$ is enough to guarantee that $\mathbb{E}[Y_N]$ tends to infinity as n grows. Conversely, when all parameters are fixed, as N grows $\mathbb{E}[Y_N]$ tends to zero, *i.e.*, sooner or later the key needs to be updated.

4.2.2 Potential Threats

As mentioned before, the fact that a response only involves $m < n$ indices makes it possible for the adversary to exploit significant information earned about a subset of key bits, even if she has no information at all about the others. To better explain this, we describe two sample attack scenarios.

Generalized Replay Attack. If a client is authenticated through a valid pair $(\mathbf{c}, R)$, and a challenge $\mathbf{c}^{(0)}$ such that $\pi_R(\mathbf{c}) = \pi_R(\mathbf{c}^{(0)})$ is subsequently generated, the authentication system becomes vulnerable to a replay attack since R would also be a valid response to $\mathbf{c}^{(0)}$. Precisely, there are exactly 2^{n-m} valid challenges for which the above condition is satisfied, and the attacker can easily detect this vulnerability by simply computing $H_m(\pi_R(\mathbf{c}), \pi_R(\mathbf{c}^{(0)}))$. More generally, if $R' \subset R$ is such that $\pi_{R'}(\mathbf{c}) = \pi_{R'}(\mathbf{c}^{(0)})$, it is possible for the adversary to define a new response R'' by simply adding $m - |R'|$ random indices to R'. Using R'' to respond to $\mathbf{c}^{(0)}$ results in an attack that succeeds with significantly higher probability than a standard response-guessing attack. To contrast similar *generalized* replay attacks, in Protocol 1 we assumed that the server uses a storage device $\mathcal{D}$ for logging authentication pairs, discarding a newly generated challenge $\mathbf{c}^{(0)}$ if there exists $(\mathbf{c}, R) \in \mathcal{D}$ such that $H_m(\pi_R(\mathbf{c}), \pi_R(\mathbf{c}^{(0)})) < k$. This condition, in fact, together with $k \sim m$, guarantees that the success probability of generalized replay attacks is comparable to that of a standard attack, *i.e.*, that replay-like attacks give the adversary no significant advantage. Let us notice that the practical relevance of replay attacks can be limited by making $|\mathcal{C}|$ large (*e.g.*, choosing $n \gg m$). In fact, a large $|\mathcal{C}|$ guarantees that $H_m(\pi_R(\mathbf{c}), \pi_R(\mathbf{c}^{(0)}))$ is large with high probability for randomly chosen R. Additionally, the δ-tolerance prescribed by SLAP makes even harder to forge valid responses based on previous ones.

Correlation-like Attack. Let us introduce a toy example in order to underline what information an eavesdropper adversary can obtain at single-bit level, and how this information can be exploited. We consider the following N authentication pairs based on parameters $n = 5$, $m = 3$, $k = 1$, $\delta = 0$, where re-

sponses have been represented using red boxes:

$$\begin{aligned}
\mathbf{s} &= (\ 1 \quad 0 \quad 0 \quad 1 \quad 0\ ) \\
\mathbf{c}_1 &= (\ 1 \quad 1 \quad 0 \quad 1 \quad 0\ ) \\
\mathbf{c}_2 &= (\ 1 \quad 1 \quad 0 \quad 1 \quad 1\ ) \\
\mathbf{c}_3 &= (\ 1 \quad 0 \quad 1 \quad 0 \quad 0\ ) \\
&\vdots \\
\mathbf{c}_N &= (\ 1 \quad 1 \quad 1 \quad 0 \quad 0\ )
\end{aligned}$$

Since the distance between each challenge and the key must be $k = 1$, and since each response has $m = 3$ components, the probability that a challenge and the key coincide at position $i \in R$ is $(m-k)/m = 2/3$. This means that the adversary can mount a *correlation-like* attack to guess key bits, and the success rate of the attack grows with the number N of eavesdropped authentication pairs. In the example, the leftmost bit of each challenge is 1 and all (shown) responses include index 0, thus suggesting to the attacker that the first key bit is also 1 (which is indeed true). Significantly less information is revealed about other indices. In general, this can happen either because an index has been used too rarely, so that the statistical sample is inadequate, or because the corresponding bit is (almost) equally often 0 or 1 in the available challenges. As a consequence, a similar attack can be prevented either (again) imposing a minimal distance among challenges over response indices, or setting $k = m/2$ (as we always do in Section 5).

4.2.3 Securing SLAP

Although we suggested specific countermeasures to the two threats presented in Section 4.2.2, we need to identify the common traits of eavesdropping attacks, and to identify a likewise general defence mechanism. The examples show that an eavesdropper adversary can succeed at exploiting the correlation between multiple sequences of indices that appear together in different valid responses. As a consequence, not only the overall entropy of the key, but also the single-bit level entropy must be kept large. This goal is achieved by imposing parameter conditions that aim at limiting superimpositions between different responses during the protocol execution. In particular, we estimate the number of repetitions of a given sequence of h bits in different responses, and we use this instrument to identify a choice of system parameters that make harder to exploit the correlation among bits of the eavesdropped authentication pairs.

Let $\{i_1, \ldots, i_h\} \subseteq [n]$ be a set of h indices. We define the random variable $C_{i_1,\ldots,i_h} = \sum_{j=1}^{N} D^{(j)}_{i_1,\ldots,i_h}$ where

$$D^{(j)}_{i_1,\ldots,i_h} = \begin{cases} 1, \text{ if } i_1, \ldots, i_h \subseteq R_j \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

$C_{i_1,\ldots,i_h}$ counts the number of times indices $i_1, \ldots, i_h \in [n]$ appear simultaneously in the valid responses $R_1, \ldots, R_N$. We can estimate this variable as follows.

Theorem 5. *In the eavesdropper adversary model, it holds*

$$\mathbb{E}[C_{i_1,\ldots,i_h}] = \frac{N}{|\mathcal{C}|} \sum_{i=k+\delta}^{n-m+k-\delta} \binom{n}{i} p_i \tag{9}$$

where

$$p_i \leq \frac{\min\left\{ \sum_{j=k-\delta}^{k+\delta} \binom{i-l}{j-l} \binom{n-i-(h-l)}{m-j-(h-l)} : l = 0, \ldots h \right\}}{\sum_{j=k-\delta}^{k+\delta} \binom{i}{j} \binom{n-i}{m-j}}. \tag{10}$$

5 PERFORMANCE EVALUATION

The main purpose of this section is to numerically evaluate the theoretical results from Section 4, in order to show that SLAP is secure against unauthorized authentication attempts.

5.1 Experimental Results

We set $m = 2k$ and $n \gg m$, accordingly to our security analysis (see Section 4.2.2), describing three main instances: *i)* the logarithmic case $m, k \in \Theta(\log_2 n)$, *ii)* the polynomial case $m, k \in \Theta(\sqrt{n})$, and *iii)* the linear case $m, k \in \Theta(n)$. We also evaluate the impact of little variations of δ.

Number of Challenges. Theorem 1 gives the number of challenges. Since $m = 2k$, we can simplify for easy evaluations

$$|\mathcal{C}| = \sum_{i=k+\delta}^{n-(m-(k-\delta))} \binom{n}{i} = \sum_{i=k+\delta}^{n-(k+\delta)} \binom{n}{i}$$

Due to binomial coefficient properties, we obtain

$$|\mathcal{C}| \geq \sum_{i=k+\delta}^{n-(k+\delta)} \binom{n}{k+\delta} \geq \left(\frac{n}{k+\delta}\right)^{k+\delta}$$

and, for $\delta = 0$, we have the rough but simple lower-bound

$$|\mathcal{C}| \geq \left(\frac{n}{k}\right)^{k} \tag{11}$$

This bound suffices to show that the number of challenges is large enough, as illustrated in Figure 1 that comprises cases where n grows exponentially, polynomially or linearly faster than k.

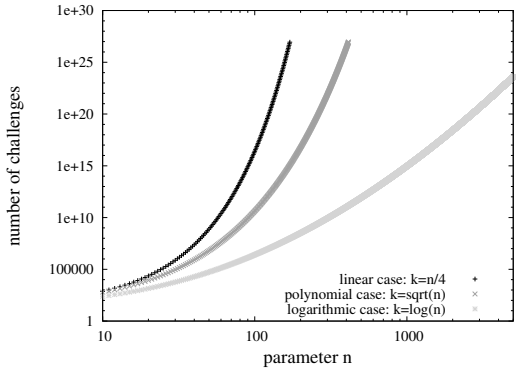

Figure 1: Number of challenges, accordingly to (11).

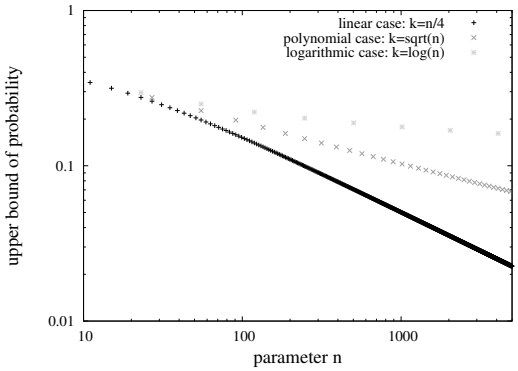

Figure 2: Probabilities of guessing a valid response in the oblivious adversary model, accordingly to (13).

Number of Valid Responses. Theorem 2 gives the number of valid responses. In particular, (3) gives the number of valid responses for a fixed distance i between challenge and secret key. To easy computations, we derive the following upper bound for $R(i)$ by applying binomial coefficient properties:

$$R(i) \leq \sum_{j=k-\delta}^{k+\delta} \binom{n/2}{j} \binom{n/2}{2k-j} \leq (2\delta+1)\binom{n/2}{k}^2 \tag{12}$$

Equation (12) shows that the number of valid responses is relatively high and, consequently, that the probability of guessing a valid response is a threat to consider. This motivates the adoption of several challenge-response cycles per authentication, as explained later in this section.

Oblivious Adversary Model. Theorem 3, especially (6), estimates the probability of randomly guessing a valid response. The estimation is evaluated through the following bound (note that $R(i)$ is maximized by choosing $i = n/2$, for $m = 2k$):

$$P(A) \leq \frac{R(n/2)}{\binom{n}{2k}} \leq \frac{(2\delta+1)\binom{n/2}{k}^2}{\binom{n}{2k}} \tag{13}$$

where we used (12) in the last passage. Figure 2 illustrates the upper bound in (13) in the linear, polynomial and logarithmic cases when $\delta = 0$. The convergence to zero is rather slow in each case, and this is mainly due to the high number of valid responses, as already advanced by (12). To overcome this issue, the server can ask the client to provide an answer to a number $h > 1$ of challenges for each authentication. If $P(A)$ is the probability of guessing a valid response for determinate parameters, then $P(A)^h < P(A)$ becomes the probability of guessing h challenges. This techniques helps at making arbitrarily harder for the attacker to obtain an unauthorized access through response-guessing attacks. The trade-off is that the parameter N, used in Section 4 to represent the number of authentications must be divided by h. Thus, reducing the total number of secure authentications.

Eavesdropper Adversary Model. Theorem 4 and Theorem 5 estimate the number of keys compatible with N authentication pairs and the repetitions of indices over N responses, respectively. The number of compatible keys, given by (7), after $N = 100$ is illustrated in Figure 3, that focuses on highlighting differences in linear, polynomial and logarithmic cases, and in Figure 4, that focuses on the parameter δ. In particular, we observe that the larger n is with respect to k (*e.g.* logarithmic and polynomial cases), the larger is the number of compatible keys; further, as n overtakes N, the number of compatible keys begins to grow exponentially faster, and δ fluctuations positively impact this growth.

For any sequence of indices $i_1, \ldots, i_h$, Theorem 5, especially (9), estimates its number of occurrences in valid responses after N authentications. The lower this estimated value is, the more unlikely it is that the attacker deploys a correlation attack. Since for $h \geq 2$ we have $\mathbb{E}[C_{i_1,i_2}] \leq \mathbb{E}[C_{i_1,\ldots,i_h}]$, we can limit ourself to analyse the case $h = 2$. We computed the upper bound derivable from (10) for $\delta = 0, 1, 5, 15$, $N = 100$, considering the linear case in Figure 5, and the polynomial case in Figure 6. We observe that the parameter δ does not affect significantly the value, and that the number of repetitions is overall low with an evident advantage in the polynomial case.

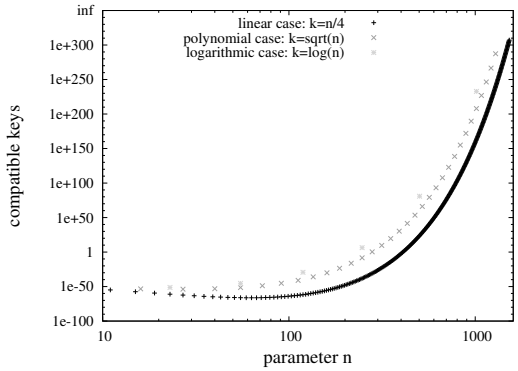

Figure 3: Number of compatible keys in the eavesdropper adversary model, accordingly to (7): $\delta = 0$, $N = 100$, linear, polynomial and logarithmic cases.

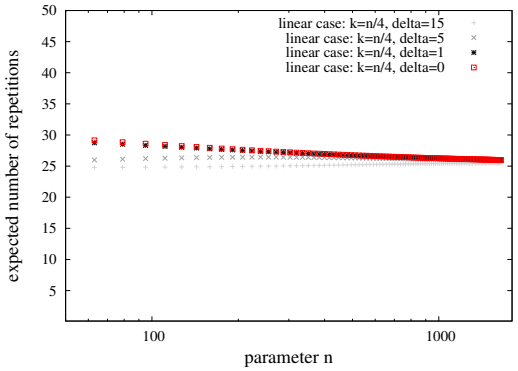

Figure 5: Repetitions of index pairs in the eavesdropper adversary model, accordingly to Theorem 5: $N = 100$, $\delta = 0, 1, 5, 15$, linear case.

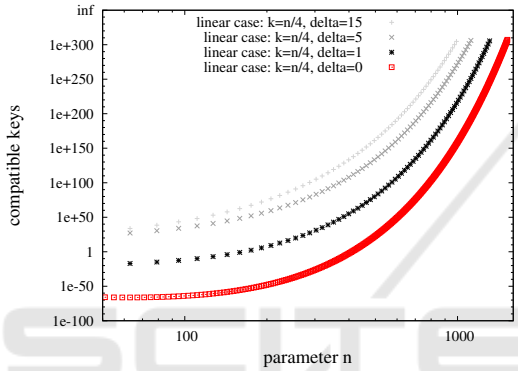

Figure 4: Number of compatible keys in the eavesdropper adversary model, accordingly to (7): $\delta = 0, 1, 5, 15$, $N = 100$, linear case.

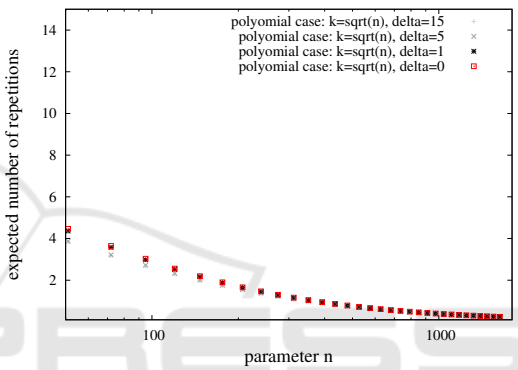

Figure 6: Repetitions of index pairs in the eavesdropper adversary model, accordingly to Theorem 5: $N = 100$, $\delta = 0, 1, 5, 15$, polynomial case.

5.2 Use Cases

We provide practical examples for specific protocol parameters, comparing results with the one-time pad based authentication scheme.

One-time Pad Based Scheme. We assume that the secret key has length $n = 1024$ and that $\varepsilon = 2^{-64}$ is the security parameter. The one-time pad based scheme can guarantee $N = 1024/64 = 16$ secure authentications, *i.e.* 64 bits per authentication.

SLAP: Linear Case. We make the above assumptions $n = 1024$ and $\varepsilon = 2^{-64}$, and we fix the parameters $(n, m, k, \delta) = (1024, 512, 256, 2)$. The number of challenges is $|\mathcal{C}| \approx 1.8 \times 10^{308}$ and the probability of randomly guessing a valid response without eavesdropping the communication is $P(A) \approx 0.174$. We can impose 26 challenge-response cycles as requirement, obtaining the wanted security: $P(A)^{26} \approx \varepsilon$. After $N = 14 \times 26 = 364$ challenge-response cycles, the probability of guessing the secret key in the eavesdropper model is $1/\mathbb{E}[Y_{416}] < \varepsilon$. For $N = 15 \times 26 = 390$ the probability is higher than ε; thus, a total of 15 authentications can be performed with security ε achieving a performance comparable with the one-time pad based authentication scheme.

SLAP: Polynomial Case. We change the parameters to $(n, m, k, \delta) = (1024, 62, 32, 1)$. The number of challenges is still approximated by $|\mathcal{C}| \approx 1.8 \times 10^{308}$ and $P(A) \approx 0.292$. Thus, 36 challenges for authentication are required to obtain $P(A)^{36} \approx \varepsilon$. We obtain that $N = 15 \times 36 = 540$ authentication pairs can be generated without compromising the security of the system, namely $1/\mathbb{E}[Y_{540}] \approx \varepsilon$. After the 16-th authentication, we obtain that $1/\mathbb{E}[Y_{576}] \approx 0.46$. This means that after 16 authentications the security of the system is compromised, as in the one-time pad based authentication. The advantage of the polynomial case over the linear case of SLAP is that repetitions of indices are less frequent (see Figure 6).

SLAP: Logarithimc Case. We change the parameters to $(n,m,k,\delta) = (1024,10,5,1)$. The number of challenges is still not varying, while $P(A) \approx 0.656$. Thus, about 105 challenges are required to obtain $P(A)^{105} \approx \varepsilon$. The system starts to be compromised after $N = 16 \times 105$ authentications, similarly to other cases and repetitions of indices are even less frequent than the polynomial case.

5.3 Discussion

The security guaranteed by our scheme is close to the theoretical limit achievable since it trades memory requirements with security properties at almost the same rate of the one-time pad based scheme. However, there a few significant differences that makes SLAP preferable for some applications. The main one is that, at each authentication, the one-time pad based scheme uses $-\log_2(\varepsilon)$ bits of the secret key while SLAP spreads the information required to be authenticated over all the n bits of the secret key. This property makes SLAP more resistant against leakages of the secret key. In fact, let us assume that the attacker breaches in the authentication server and successfully recovers few bits, *i.e.* $-\log_2(\varepsilon)$, of the secret key. If the server implements the one-time pad based scheme, then the attacker can leak the bits required for the next authentication gaining access to the system. If the server implements SLAP, the attacker has not earned enough information to generate valid responses. In fact, to generate a response for a given challenge, the attacker must know at least $k \pm \delta$ bits of the secret key where the challenge differs and $m-(k\pm\delta)$ where it coincides. Hence, at least m bits of the secret key must be leaked by the attacker to generate a valid response to one challenge (and it is still not enough to be authenticated). This property makes SLAP more resistant than the one-time based scheme against leakages of the secret key.

Another difference is that SLAP is more resistant against successful impersonation attacks. In fact, let us assume that a malicious server (*i.e.* an attacker personifying the server) collects answers from a client that wants to be authenticated. If the one-time pad based scheme is the algorithm of choice, then the answers generated by the client are the bits of the secret key; thus, the malicious server gains access to the system by using these bits. Instead, with SLAP, the answers generated by the client are valid responses for particular challenges. This means that the malicious server has to ask for a challenge to the real server, and forward it to the client; this step introduces a detectable delay in the process since multiple challenge-response pairs are required for one authentication.

The last difference is the flexibility offered by SLAP. Adopting the SLAP protocol, the server has the capability of varying the number of challenges required to be authenticated in the system *during* the execution of the protocol – *i.e.* increasing or decreasing the number of possible secure authentications. For instance, it might be the case that a server features different services but that not all of them needs to be protected with the highest level of security. Using SLAP, the server has the flexibility of setting different security requirements for each of these services by simply defining, for each of them, the number of challenges that need to be answered. The server can also change these values at any time during the execution of the protocol without acknowledging clients; indeed, clients do not necessarily need to know how many answers they must provide for gaining the access to a particular service. In the one-time pad based scheme it is certainly possible to subdivide the secret key to achieve the same goal, but clients must be aware of this subdivision as well and the management effort of the server becomes more complicated.

6 CONCLUSION

In this paper we have introduced SLAP, a light-weight authentication protocol that guarantees a predictable number of secure authentications. The number of authentication rounds is proportional to the length of the secret key shared between server and client, and thus it is customizable. Differently from prior work, the proposed protocol is only based on information-theoretic security methods, thus attackers with strong computational capabilities have no advantages. The simplicity of the SLAP scheme makes it an ideal authentication protocol to be implemented in RFID tags, IoT scenarios, or other contexts where light-weight protocols are required.

The thorough mathematical analysis of SLAP's features has been supported by an extensive experimental campaign – run with real world system parameters. Achieved results show the efficacy and practicality of the proposed solution. To further assess the concrete impact of SLAP, in the next future we plan to both pinpoint SLAP's theoretical implant, and carefully assess its resource footprint.

REFERENCES

Alippi, C., Bogdanov, A., and Regazzoni, F. (2014). Lightweight cryptography for constrained devices. In

Integrated Circuits (ISIC), 2014 14th International Symposium on, pages 144–147.

Armknecht, F., Hamann, M., and Mikhalev, V. (2014). Lightweight authentication protocols on ultra-constrained rfids - myths and facts. In *Radio Frequency Identification: Security and Privacy Issues - 10th International Workshop, RFIDSec 2014, Oxford, UK, July 21-23, 2014, Revised Selected Papers*, pages 1–18.

Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805.

Aumasson, J.-P., Henzen, L., Meier, W., and Naya-Plasencia, M. (2010). Quark: A lightweight hash. In *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 1–15. Springer.

Bernstein, D. J. and Lange, T. (2012). Never trust a bunny. In Hoepman, J.-H. and Verbauwhede, I., editors, *RFIDSec*, volume 7739 of *Lecture Notes in Computer Science*, pages 137–148. Springer.

Bowers, K. D., Juels, A., Rivest, R. L., and Shen, E. (2013). Drifting keys: Impersonation detection for constrained devices. In *INFOCOM*, pages 1025–1033. IEEE.

Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and Wright, S. J. (2008). Power awareness in network design and routing. In *INFOCOM*, pages 457–465. IEEE.

Engels, D. W., Fan, X., Gong, G., Hu, H., and Smith, E. M. (2010). Hummingbird: Ultra-lightweight cryptography for resource-constrained devices. In *Financial Cryptography Workshops*, volume 6054 of *Lecture Notes in Computer Science*, pages 3–18. Springer.

Engels, D. W., Saarinen, M.-J. O., Schweitzer, P., and Smith, E. M. (2011). The hummingbird-2 lightweight authenticated encryption algorithm. *IACR Cryptology ePrint Archive*, 2011:126.

Feldhofer, M., Wolkerstorfer, J., and Rijmen, V. (2005). Aes implementation on a grain of sand. *IEE Proceedings-Information Security*, 152(1):13–20.

Gaspar, L., Leurent, G., and Standaert, F. (2014). Hardware implementation and side-channel analysis of lapin. In *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, pages 206–226.

Gilbert, H., Robshaw, M. J. B., and Seurin, Y. (2008). Hb#: Increasing the security and efficiency of hb+. In Smart, N. P., editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

Heyse, S., Kiltz, E., Lyubashevsky, V., Paar, C., and Pietrzak, K. (2012). Lapin: An efficient authentication protocol based on ring-lpn. In Canteaut, A., editor, *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 346–365. Springer.

Hopper, N. J. and Blum, M. (2001). Secure human identification protocols. In Boyd, C., editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer.

Juels, A. and Weis, S. A. (2005). Authenticating pervasive devices with human protocols. In *CRYPTO*, pages 293–308.

Kao, Y.-W., Luo, G.-H., Lin, H.-T., Huang, Y.-K., and Yuan, S.-M. (2011). Physical access control based on qr code. In *Cyber-enabled distributed computing and knowledge discovery (CyberC), 2011 International Conference on*, pages 285–288. IEEE.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397.

Kocher, P. C. (1996). Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Koblitz, N., editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer.

Oder, T., Pöppelmann, T., and Güneysu, T. (2014). Beyond ECDSA and RSA: lattice-based digital signatures on constrained devices. In *The 51st Annual Design Automation Conference 2014, DAC '14, San Francisco, CA, USA, June 1-5, 2014*, pages 1–6.

Patton, E. W. and McGuinness, D. L. (2014). A power consumption benchmark for reasoners on mobile devices. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 409–424.

Pietrzak, K. (2012). Cryptography from learning parity with noise. In *SOFSEM*, volume 7147 of *Lecture Notes in Computer Science*, pages 99–114. Springer.

Raza, S., Shafagh, H., Hewage, K., Hummen, R., and Voigt, T. (2013). Lithe: Lightweight Secure CoAP for the Internet of Things. *Sensors Journal, IEEE*, 13(10):3711–3720.

RSA, The Security Division of EMC (2015). RSA SecureId token. Available on line.

Sehgal, A., Perelman, V., Kuryla, S., and Schönwälder, J. (2012). Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, 50(12):144–149.

Shannon, C. (1949). Communication theory of secrecy systems. *Bell System Technical Journal, Vol 28, pp. 656-715*.

Vogt, H. (2002). Efficient object identification with passive RFID tags. In *Proceedings of the First International Conference on Pervasive Computing*, volume 2414 of *Lecture Notes in Computer Science*, pages 98–113, Zurich. Springer-Verlag.

Weir, C. S., Douglas, G., Carruthers, M., and Jack, M. A. (2009). User perceptions of security, convenience and usability for ebanking authentication tokens. *Computers & Security*, 28(1-2):47–62.

APPENDIX

Let $\mathcal{R} = \{R \subset [n], |R| = m\}$ be the set of *formally valid* responses. Our envisaged adversaries can be modelled as follows.

Definition 1 (Oblivious Adversary Model)**.** *We define the discrete probability space* $(\Omega, \mathcal{F}, P)$ *where*

i) The sample space is

$$\Omega = \{(\mathbf{s}, \mathbf{c}, R) \in \mathbb{F}_2^n \times \mathcal{C} \times \mathcal{R}\};$$

ii) $\mathcal{F} = 2^\Omega$ *is the set of events (each event is a subset of outcomes);*

iii) $P : \Omega \to [0,1]$, $E \subseteq \Omega \mapsto \frac{|E|}{\Omega}$ *is the probability mass function that assigns events to probabilities.*

Definition 2 (Eavesdropper Adversary Model)**.** *We define the discrete probability space* $(\Omega, \mathcal{F}, P)$ *where*

i) the sample space is

$$\Omega = \Big\{\{\mathbf{s}, (\mathbf{c}_1, R_1), \ldots, (\mathbf{c}_N, R_N)\} \in \mathbb{F}_2^N \times (\mathcal{C} \times \mathcal{R})^N s.t. \; H_m(\pi_{R_i}(\mathbf{s}), \pi_{R_i}(\mathbf{c}_i)) \in [k-\delta, k+\delta], \forall i = 1, \ldots, N\Big\};$$

ii) $\mathcal{F} = 2^\Omega$ *is the set of events;*

iii) $P : \Omega \to [0,1]$, $E \subseteq \Omega \mapsto \frac{|E|}{\Omega}$ *is the probability mass function that assigns events to probabilities.*

Proof of Theorem 1. Given a challenge $\mathbf{c} \in \mathcal{C}$, a valid response for $\mathbf{c}$ is a subset $R \subset [n]$ of m indices such that $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) \in [k-\delta, k+\delta]$, where $\mathbf{s} \in \mathbb{F}_2^n$ is the secret key selected by the protocol. To generate a valid response, the secret key $\mathbf{s}$ and the challenge $\mathbf{c}$ must differ in, at least, $j \in [k-\delta, k+\delta]$ of their components, while coinciding in $m-j$ components. This is equivalent to

$$\exists j \in [k-\delta, k+\delta] : \; j \leq H_n(\mathbf{s}, \mathbf{c}) \leq n - (m-j). \quad (14)$$

When Eq. (1) is satisfied, then also Eq. (14) is satisfied by all $j \in [k-\delta, k+\delta]$. Hence, for all values of j in this range, the challenge $\mathbf{c}$ admits a valid response R such that $H_m(\pi_R(\mathbf{s}), \pi_R(\mathbf{c})) = j$. Then, for any distance $i \in \{k+\delta, \ldots, n-(m-(k-\delta))\}$, there are $\binom{n}{i}$ combinations where exactly i components of $\mathbf{s}$ are left unchanged. Summing over i, we obtain the cardinality of $\mathcal{C}$ and Eq. (2). □

Proof of Theorem 2. For a fixed Hamming distance i, to generate a valid response for $\mathbf{c}$, the client chooses any number $j \in [k-\delta, k+\delta]$, j out of i components where $\mathbf{s}$ and $\mathbf{c}$ differ, and $m-j$ out of $n-(m-j)$ components where they coincide. Summing all the combinations over j, we obtain Eq. (3). Due to Eq. (1), the admitted values of the Hamming distance between $\mathbf{s}$ and $\mathbf{c}$ are $\{k+\delta, \ldots, n-(m-(k-\delta))\}$. Thus, Eq. (4) is obtained by summing the number of valid responses for any admitted value of distance. □

Proof of Lemma 1. By definition of conditional probability, we have

$$P\big(A \mid \{\mathbf{s} = \mathbf{s}^{(0)}, \mathbf{c} = \mathbf{c}^{(0)}\}\big) = \frac{P\big(A \cap \{\mathbf{s} = \mathbf{s}^{(0)}\} \cap \{\mathbf{c} = \mathbf{c}^{(0)}\}\big)}{P\big(\{\mathbf{s} = \mathbf{s}^{(0)}\} \cap \{\mathbf{c} = \mathbf{c}^{(0)}\}\big)}.$$

Due to Eq. (3) of Theorem 2 and iii) of Definition 1, the numerator is

$$\frac{R\big(H_n(\mathbf{s}^{(0)}, \mathbf{c}^{(0)})\big)}{|\Omega|} = \frac{R(j)}{|\Omega|}$$

since this ratio coincides with the number of favourable outcomes divided by the total number of outcomes. Instead, the denominator is $\binom{n}{m}/|\Omega|$ since there are $\binom{n}{m}$ (*i.e.* number of possible responses) favourable outcomes. Assembling together numerator and denominator, we conclude the proof. □

Proof of Theorem 3. $P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}\})$ by definition of conditional probability is equivalent to

$$\sum_{\mathbf{c}^{(0)} \in \mathcal{C}} P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}, \mathbf{c} = \mathbf{c}^{(0)}\}) P(\{\mathbf{c} = \mathbf{c}^{(0)}\}).$$

Due to Eq. (5), the sum can be rewritten as

$$\sum_{\mathbf{c}^{(0)} \in \mathcal{C}} \frac{R(H_n(\mathbf{s}^{(0)}, \mathbf{c}^{(0)}))}{\binom{n}{m}} \frac{1}{|\mathcal{C}|} = \frac{1}{|\mathcal{C}|\binom{n}{m}} \sum_{\mathbf{c}^{(0)} \in \mathcal{C}} R(H_n(\mathbf{s}^{(0)}, \mathbf{c}^{(0)})).$$

Thus, by rearranging terms in the sum, we obtain

$$P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}\}) = \frac{1}{|\mathcal{C}|\binom{n}{m}} \sum_{j=k+\delta}^{n-(m-(k-\delta))} \binom{n}{j} R(j). \quad (15)$$

To compute $P(Aut)$, we use again the definition of conditional probability to conclude the proof

$$\begin{aligned} P(Aut) &= \sum_{\mathbf{s}^{(0)} \in \mathbb{F}_2^n} P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}\}) P(\{\mathbf{s} = \mathbf{s}^{(0)}\}) = \\ &= \sum_{\mathbf{s}^{(0)} \in \mathbb{F}_2^n} P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}\}) 2^{-n} = \\ &= P(Aut \mid \{\mathbf{s} = \mathbf{s}^{(0)}\}). \end{aligned}$$

□

Proof of Theorem 4. We start by enumerating each vector of n components, namely by writing $\mathbb{F}_2^n = \{\mathbf{v}^{(i)}, \forall i = 1, \ldots, 2^n\}$. This allows us to introduce the random variables

$$X_N^{(i)} : \Omega \longrightarrow \mathbb{N}, \; \omega \mapsto \begin{cases} 1, \text{ if } \mathbf{v}^{(i)} \in S_N(\omega) \\ 0, \text{ otherwise} \end{cases}$$

and to write Y_N as a sum of them, namely

$$Y_N = \sum_{i=1}^{2^n} X_N^{(i)}. \tag{16}$$

Due to the linearity of the expected value, we have

$$\mathbb{E}[Y_N] = \sum_{i=1}^{2^n} \mathbb{E}[X_N^{(i)}]. \tag{17}$$

Proving the following three facts will conclude the proof:

i) $\mathbb{E}[X_N^{(i)}] = \mathbb{E}[X_N^{(j)}], \forall i, j$;
ii) $\mathbb{E}[X_N^{(i)}] = \mathbb{E}[X_1^{(i)}]^N$;
iii) $\mathbb{E}[X_1^{(i)}] = \rho = 2^{-m} \sum_{j=k-\delta}^{k+\delta} \binom{m}{j}$.

Claim *i*) means that, from the attacker perspective, each vector in $\mathbb{F}_2^n$ is equally likely to belong to S_N. This is true because, even if a vector $\mathbf{v}^{(i)}$ is more likely to belong to S_N for a particular outcome ω_i, the sample space Ω allows both the secret key $\mathbf{s}$ and the set I_N to vary among all possible combinations in Ω; thus, any other vector $\mathbf{v}^{(j)}$ would also be more likely to belong to S_N for the appropriate outcome ω_j. Differently said, the symmetries in the probabilistic model imply that, on average, all the vectors are equally likely of being in S_N.

To prove claim *ii*), we start writing the event $\{\omega \in \Omega : \mathbf{v}^{(i)} \in S_N(\omega)\}$ as intersection of the sets $A_j = \{\omega \in \Omega : H_m(\pi_{R_j}(\mathbf{v}^{(i)}), \pi_R(\mathbf{c}_j)) \in [k-\delta, k+\delta]\}$ representing the outcomes in which $\mathbf{v}^{(i)}$ is compatible with the j-th authentication pair. Hence, we have

$$\mathbb{E}[X_N^{(i)}] = P\big(\{\omega \in \Omega : \mathbf{v}^{(i)} \in S_N(\omega)\}\big) = P\big(\bigcap_{j=1}^{N} A_j\big).$$

Since the authentication pairs $(\mathbf{c}_j, R_j)$ are uniformly at random chosen by the protocol, the events A_j are independent from each other and we can write

$$\mathbb{E}[X_N^{(i)}] = P\big(\bigcap_{j=1}^{N} A_j\big) = \prod_{j=1}^{N} P(A_j) = \prod_{j=1}^{N} \mathbb{E}[X_1^{(i)}] = \mathbb{E}[X_1^{(i)}]^N.$$

Claim *iii*) follows from next considerations. For any secret key $\mathbf{s}$ and for any N, when the authentication pair $(\mathbf{c}, R)$ is used for an authentication, the attacker acquires the information that a number $j \in [k-\delta, k+\delta]$ of bits of the challenge $\mathbf{c}$, among those m indexed by R, differ from bits of $\mathbf{s}$. These j bits can be combined in $\binom{m}{j}$ different ways, and j can be any value in $[k-\delta, k+\delta]$. We notice also that the attacker is not earning information on the remaining $n-m$ bits of $\mathbf{s}$. Thus, by eavesdropping a single authentication pair, the attacker is equally uncertain on a total of

$$2^{n-m} \sum_{j=k-\delta}^{k+\delta} \binom{m}{j}$$

possible secret keys, that is the cardinality of S_1. In particular, we have,

$$\mathbb{E}[X_1^{(i)}] = P\big(\{\mathbf{v}^{(i)} \in S_1\}\big) = \frac{2^{n-m} \sum_{j=k-\delta}^{k+\delta} \binom{m}{j}}{2^n} = \rho.$$

□

Proof of Theorem 5. Since challenges and responses are independently at random chosen,

$$\mathbb{E}[C_{i_1,\ldots,i_h}] = \sum_{j=1}^{N} \mathbb{E}[D_{i_1,\ldots,i_h}^{(j)}] = N\mathbb{E}[D_{i_1,\ldots,i_h}^{(j)}],$$

where the last passage is justified by the fact that challenges and responses are also identically distributed.

For a given challenge $\mathbf{c}$ at distance i from the secret key, accordingly to Eq. (3), there is a number

$$R(i) = \sum_{j=k-\delta}^{k+\delta} \binom{i}{j} \binom{n-i}{m-j}$$

of valid responses but only a fraction $R(i,h)$ of them contains the sequence $i_1, \ldots, i_h$. Since the sequence $i_1, \ldots, i_h$ is fixed, we can define $l \in \{0, \ldots, h\}$ as the number of indices in $\{i_1, \ldots, i_h\}$ where $\mathbf{c}$ differs from the secret key. In this way, $R(i,h)$ can be written in function of l:

$$R(i,h) = R(i,h,l) = \sum_{j=k-\delta}^{k+\delta} \binom{i-l}{j-l} \binom{n-i-(h-l)}{m-j-(h-l)}.$$

In particular, we have the inequality

$$R(i,h) \leq \min\{R(i,h,l) : l = 0, \ldots h\}.$$

Hence, for a challenge at distance i from the secret key, the probability that indices $i_1, \ldots, i_h$ appear together in a valid response is bounded by

$$p_i = \frac{R(i,h)}{R(i)} \leq \frac{\min\{R(i,h,l) : l \in \{0, \ldots, h\}\}}{R(i)}.$$

The probability that a uniformly at random selected challenge has distance i from the secret key is $\binom{n}{i}/|C|$. Thus,

$$\mathbb{E}[D_{i_1,\ldots,i_h}^{(j)}] = \sum_{i=k+\delta}^{n-(m-(k-\delta))} \frac{\binom{n}{i}}{|C|} p_i$$

and, multiplying this value by N, we conclude the proof. □