Breaking and Fixing the HB+DB protocol

Ioana Boureanu[1], David Gerault[2], Pascal Lafourcade[2], and Cristina Onete[3]

[1] University of Surrey SCCS i.boureanu@surrey.ac.uk
[2] University Clermont Auvergne LIMOS david.gerault@uca.fr, pascal.lafourcade@uca.fr
[3] Université de Rennes 1/IRISA c.onete@gmail.com

Abstract. The HB protocol and its HB$^+$ successor are lightweight authentication schemes based on the Learning Parity with Noise (LPN) problem. They both suffer from the so-called GRS-attack whereby a man-in-the-middle (MiM) adversary can recover the secret key. At WiSec 2015, Pagnin *et al.* proposed the *HB+DB* protocol: HB$^+$ with an additional distance-bounding dimension added to detect and counteract such MiM attacks. They showed experimentally that HB+DB was resistant to GRS adversaries, and also advanced HB+DB as a distance-bounding protocol, discussing its resistance to worst-case distance-bounding attackers.

In this paper, we exhibit flaws both in the authentication and distance-bounding layers of HB+DB; these vulnerabilities encompass practical attacks as well as provable security shortcomings. First, we show that HB+DB may be impractical as a secure distance-bounding protocol, as its distance-fraud and mafia-fraud security-levels scale poorly compared to other distance-bounding protocols. Secondly, we describe an effective MiM attack against HB+DB: our attack refines the GRS-strategy and still leads to key-recovery by the attacker, yet this is *not* deterred by HB+DB's distance-bounding. Thirdly, we refute the claim that HB+DB's security against passive attackers relies on the hardness of the LPN problem. We also discuss how (erroneously) requiring such hardness, in fact, lowers HB+DB's efficiency and its resistance to authentication and distance-bounding attacks.

Drawing on HB+DB's design flaws, we also propose a new distance-bounding protocol – $\mathbb{BLOG}$. It retains parts of HB+DB, yet $\mathbb{BLOG}$ is provably secure, even – in particular – against MiM attacks. Moreover, $\mathbb{BLOG}$ enjoys better practical security (asymptotical in the security parameter).

1 Introduction

Providing secure authentication and identification represents an important goal of modern cryptography. Authentication not only enables efficient access-control to resources such as sensitive areas or privileges, but it also constitutes a building-block used towards more comprehensive cryptographic guarantees, like end-to-end confidentiality and integrity. In authentication protocols, a device typically called a *prover* (*i.e.*, an electronic passport or an RFID parking pass) must prove its legitimacy to a *verifier*, usually a reader. The protocol is *correct* if a legitimate prover is (almost) always authenticated. And, an authentication protocol is *secure* if it guarantees *impersonation security*: a man-in-the-middle (MiM) attacker must have a negligible probability to succeed in being authenticated as a legitimate prover.

The HB protocol family. The *HB* protocol [16] is arguably one of today's best-known lightweight authentication schemes. It aims to provide secure *human* authentication, *i.e.*, given limited abilities to compute and remember data. This also makes HB suitable to resource-constrained devices.

A few years later, the HB$^+$ protocols were introduced [17] for specific RFID systems. The authors have shown that, if the learning parity with noise (LPN) problem [3] is hard, the HB$^+$ protocol is impersonation-secure against *passive* attackers. However, *active* MiM attackers can successfully impersonate provers, for instance by means of the GRS attack [15]. Subsequent improvements [10,14] on HB$^+$resolve this flaw by relying on extensions of LPN [9]. In turn, these yield solutions computationally far less efficient than HB$^+$, which cannot be used in resource-constrained devices.

Distance bounding. Distance-bounding (DB) protocols [8] can be viewed as enhanced authentication schemes, which aim to deter impersonation attempts mounted via relaying. To this end, the verifier is equipped with a *clock* and measures the *roundtrip time* (RTT) during certain exchanges. If the measured RTT exceeds a given *proximity bound*, then the verifier will reject the prover regardless of the validity of its authenticating responses. In addition to the authentication-driven impersonation security, secure DB also requires (at least) the following three properties: resistance to distance-fraud, mafia-fraud, and terrorist-fraud. (See Section 2.1.).

The HB+DB scheme. The GRS attack [15] is a well-known key-learning vulnerability of HB$^+$, which allows MiM attackers to learn a long-term secret by altering messages in their transit from the verifier to the prover. (See Section 2.3.). An intuitive countermeasure to the GRS attack is to enhance HB$^+$ with a proximity-checking dimension, as in DB protocols. This was used to advance the *HB+DB* protocol in [20] and the countermeasure was evaluated by practical experiments. As such, the principal aim of HB+DB appears to be that of thwarting active impersonation attempts *in practice*, rather than providing provable impersonation security against worst-case attackers. However, the authors of HB+DB *also* discuss theoretical aspects: among others, they include some level of worst-case security analyses, and determine certain LPN-noise levels for security against worst-case passive attackers. In this paper, we present failings of HB+DB, which are linked to practice (*e.g.*, resistance to distance-fraud and mafia-fraud scaling poorly) as well as to provable security (*e.g.*, HB+DB's security against passive attacker not relying on LPN-hardness, despite claims to the contrary [20].).

Our contributions. Our results are as follows.

- **DB security in HB+DB.** Pagnin *et al.* specifically require LPN-hardness for the security of HB+DB (see [21], p.11). Thus, the LPN-noise probability should be of $\frac{1}{8}$ or $\frac{1}{4}$ [19]. Such noise factors were tacitly included in HB+DB's distance-fraud analysis [21], though not explicitly explored in the context of HB+DB security.
  In Section 3.1, we show how such noise ratios impact HB+DB's false-acceptance rates, yielding highly-successful distance- and mafia-fraud attacks even for when HB+DB would use a large number of rounds.
- **Active MiM insecurity in HB+DB.** Section 3.2 shows how to retrieve the long-term key used in HB+DB by a MiM attacks slightly different to the GRS attack. Instead of *flipping* bits (which requires time-consuming demodulation), our attacker exploits its knowledge of bit-durations and uses amplification to *inflict* a specific value to a particular bit of the challenge. Eventually, by repeating this strategy, the attacker retrieves the long-term key. This approach is faster than the GRS attack, and cannot be ruled out via experiments à la Pagnin *et al.* This suggests that proposing a protocol that attains *provable* security may be a better way to achieve active-MiM resistance even in practice.
- **Poor Scaling of Security vs. Robustness in HB+DB.** Section 3.2 also includes an asymptotic study of HB+DB's parameters, in the light of the active attack we put forward. This shows a poor scaling of security vs. robustness in HB+DB. For instance, even for a very large key-size of 2048 bits, HB+DB can achieve a maximum of 26 bits of security (against impersonation or key-recovery), if HB+DB is to reject no more than 1% of legitimate authentication attempts.

- **HB+DB and LPN-hardness.** Section 3.4 shows that, despite claims to the contrary [21], HB+DB's security against passive attackers no longer reduces to the hardness of the underlying LPN instance. This is due to the HB+DB's design obscuring an "LPN coefficient", which is no longer visible to the passive attacker.
- **Fixing HB+DB.** Drawing on the above, we propose a DB protocol: $\mathbb{BLOG}$. To run as close as possible to HB+DB, we maintain HB+DB's k-bit long challenges, inheriting this unsuitability for use in lightweight devices. However, we explicitly decouple $\mathbb{BLOG}$'s security from LPN and we bank on more commonplace constructions, *e.g.*, PRFs; this design-choice is due LPN-driven noise-levels rendering DB protocols less secure (as showed herein). To this end, unlike HB+DB, our $\mathbb{BLOG}$ protocol attains near-optimal bounds for distance- and mafia-fraud. Moreover, our $\mathbb{BLOG}$ protocol offers *provable* distance-bounding security.

Firstly, we underline that composing authentication protocols, such as HB$^+$, with a distance-bounding dimension requires great care, as it may result in security vulnerabilities. For instance, authentication requirements for HB$^+$ such as high, LPN-driven noise-levels yield poor distance-bounding security. This in turn translates into HB+DB itself being an insecure distance-bounding scheme, if one follows HB+DB's authors guidelines on LPN-driven noise. Not only does HB+DB's distance-bounding security scale poorly, but –ultimately– HB+DB seems to counteract only the specific GRS attack, evaluated in HB+DB's accompanying experiments in [21]. In particular, attack strategies different from the GRS approach may well be out of the scope of the experiments in [21] and can defeat HB+DB's alleged/practical MiM security. In this paper, we indeed show that HB+DB is vulnerable to at least one such key-recovery attack.

Thus, we advocate using worst-case, provable security analyses when proposing protocol compositions. As such, we also construct a protocol similar to HB+DB that is *provably* secure.

Secondly, we note that meaningful compositions between LPN-based authentication protocols and distance-bounding are not trivial (if at all possible), due to LPN-driven noise. As we show that HB+DB is in fact not LPN-based, our construction also dispenses of the LPN-driven noise, yielding better distance-bounding security.

2 Preliminaries

2.1 Authentication and DB Security

We first describe, at an intuitive level, the security models we use for authentication and DB; we refer the reader to [12] for the formalisation.

Authentication. A symmetric-key authentication protocol is a triplet of algorithms $\textsc{Auth}$= ($\textsc{Auth}.\mathsf{KGen}$, $\textsc{Auth}.\mathsf{P}$, $\textsc{Auth}.\mathsf{Vf}$), taking as input a security parameter, such that:

- $\textsc{Auth}.\mathsf{KGen}$: on the security parameter at input, this algorithm outputs a secret key sk (consisting of one or multiple keys). For HB$^+$, $sk = (x, y)$;
- $\textsc{Auth}.\mathsf{P}$, $\textsc{Auth}.\mathsf{Vf}$: on the security parameter at input, the Prove algorithm $\textsc{Auth}.\mathsf{P}$ operates interactively with the Verify algorithm $\textsc{Auth}.\mathsf{Vf}$ to produce, on the verifier side, an authentication bit Out_V (set to 1 if the authentication succeeds, and to 0 otherwise).

Secure authentication is defined *à la* Bellare and Rogaway [2] in terms of two properties. *Correctness* demands that for all honestly-generated keys, the interaction of the $\textsc{Auth}.\mathsf{P}$ and $\textsc{Auth}.\mathsf{Vf}$ algorithms yields a verifier output equal to 1. *Impersonation security* is defined w.r.t. a man-in-the-middle (MiM) adversary, which can communicate with independent instances of the prover and with the verifier and can eavesdrop on honest prover-verifier exchanges. The protocol is impersonation insecure if such a MiM can make the verifier accept its legitimacy in a protocol-session, during which the MiM cannot interact with the prover.

Secure Distance-Bounding. In distance bounding (DB), the prover authenticates not just by proving he has the secret key sk, but also by demonstrating he is no further from the verifier than a time/distance bound t_{max}. Most DB protocols consist of multiple message exchanges called *rounds*. Rounds in which the verifier does or does not measure the Round Trip Time (RTT) are called *time-critical* and *lazy*, respectively. DB protocols generally have three stages: 1. **Initialization**. This is formed of lazy rounds, in which the prover and the verifier exchange data, *e.g.*, nonces, and also compute session-specific material to be used later. 2. **Distance-Bounding**. This is formed of time-critical rounds, whereby the prover answers challenges by the verifier. The verifier stores the responses and the measured roundtrip times. 3. **Verification**: This may be formed of additional lazy rounds. Finally, the verifier uses the responses from the time-critical rounds, the RTTs, the bound t_{max}, and potentially some further data from the lazy rounds to produce the output bit Out_V.

In addition to impersonation security, DB must also guarantees the following properties: 1. **Mafia-fraud resistance:** An active MiM attacker cannot illegitimately authenticate to the verifier even if it has access to an honest prover. 2. **Distance-fraud resistance:** A malicious prover located outside the verifier's proximity cannot successfully pass the verifier's proximity-check. 3. **Terrorist-fraud resistance:** A malicious prover cannot help a MiM attacker authenticate successfully to the verifier without allowing the adversary to authenticate arbitrarily afterwards. Notably, DB protocols also thwart *relay attacks*, consisting of the exact forwarding of messages between an honest prover and an honest verifier. Such attacks exploit two weaknesses particularly inherent to low-resource, passive devices: (i) Honest provers (usually) respond spontaneously even to an unauthenticated, possibly-malicious device, without a reactive input/consent from the prover or its holder; (ii) The verifier cannot attest the *identity* of its communication partner – it can, at most, verify the legitimacy of its messages.

Herein we use the so-called *DFKO* distance-bounding formalism [12], which is a session-based model. First, we describe the DFKO's formalisation of *mafia fraud* (MF). The MF adversary can interact with *both prover and verifier* during his attack. If a MF adversary does *pure relaying* (*i.e.*, proxying messages back and forth between a prover and a verifier), then the adversary-verifier session is said to be *tainted*. The attacker may relay messages in some sessions (*e.g.*, in order to learn a long-term secret key), but not in others. The MF adversary wins if it makes the verifier accept it as legitimate in an untainted adversary-verifier session.

In the DFKO definition of *distance fraud* (DF), an adversary who the secret key can interact with the verifier arbitrarily. In each time-critical round, the adversary must *commit* to a response before the verifier's challenge is sent. If no commitment is made, or if the adversary changes the message to which it has committed once the challenge is received, then $\mathcal{A}$ taints the adversary-verifier session. The adversary wins if it authenticates to the verifier in an untainted session.

We use the SimTF definition of terrorist-fraud from the DFKO model. The *terrorist* adversary first interacts with a malicious prover with the goal of authenticating to the honest verifier. The attacker may query the prover arbitrarily during lazy protocol rounds, but the two parties may not interact during time-critical exchanges. Finally, after the terrorist adversary wins, its entire internal state is transferred to a *simulator* $\mathcal{S}$, which must authenticate to the verifier without the prover's help. A DB protocol is *terrorist-fraud resistant* if and only if for any terrorist adversary winning with some probability $p_{\mathcal{A}}$ there exists a simulator inheriting its full internal state, which wins with probability $p_{\mathcal{S}} \geq p_{\mathcal{A}}$.

2.2 The LPN Problem

Let x be a uniformly sampled k-bit vector. Let $\eta \in (0, 1/2)$ be a *constant* noise parameter and ϵ be an n-bit vector such that its Hamming weight is smaller than $\eta \times n$, *i.e.*, $HW(\epsilon) \leq \eta \times n$. An instance $LPN_{x,\eta}$ of the LPN problem [3] involves solving the equation $r = (A \cdot x) \oplus \epsilon$ in x, given a uniformly-sampled $n \times k$ binary matrix A and the n-bit vector r produced as shown above.

For a matrix A of sub-exponential size (in the security parameter), even the best-known algorithms can solve $LPN_{x,\eta}$ only in sub-exponential time complexity [4]. This makes well-parameterised protocols from the HB family secure against polynomial-time, passive attackers, as their transcripts describe hard LPN instances.

2.3 The HB$^+$ and HB+DB protocols

HB$^+$'s Description. The HB$^+$ protocol, which is partly depicted in Fig. 1, assumes that both the prover and the verifier posses two, k-bit long shared keys x, y. At each session, the verifier and the prover generate the bitstrings a and b, respectively. The prover is authenticated by means of a response r, which relies on the two secrets (blinded respectively by a and b), and an additional *noise* term, as per Fig. 1. The noise term ϵ is selected according to a Bernoulli distribution with a public mean $\eta \in (1, \frac{1}{2})$. Typical values used for η are $\frac{1}{8}$ or $\frac{1}{4}$ [19], since lower values reveal more information about the secret keys, whereas values approaching $\frac{1}{2}$ detract from correctness, making honest provers be rejected fail. The round depicted in Fig. 1 is repeated n times. Clearly, the larger n is, the better the correctness of the protocol.

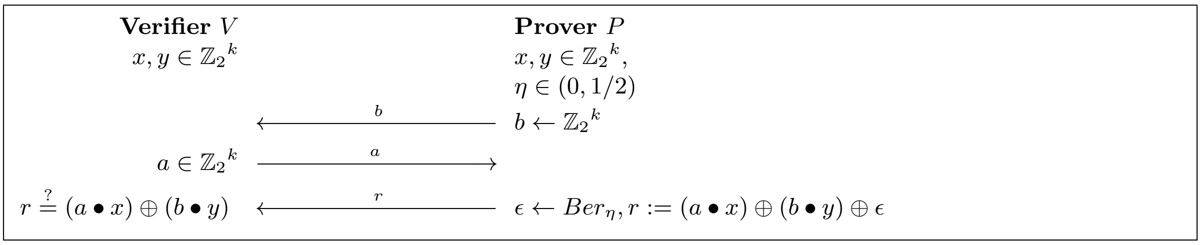

Verifier V
$x, y \in {\mathbb{Z}_2}^k$
Prover P
$x, y \in {\mathbb{Z}_2}^k$,
$\eta \in (0, 1/2)$
b
$b \leftarrow {\mathbb{Z}_2}^k$
$a \in {\mathbb{Z}_2}^k$
a
$r \overset{?}{=} (a \bullet x) \oplus (b \bullet y)$
r
$\epsilon \leftarrow Ber_\eta, r := (a \bullet x) \oplus (b \bullet y) \oplus \epsilon$

Fig. 1: One round of the HB^+ protocol.

HB$^+$'s Security. HB$^+$ guarantees only unilateral, prover to verifier authentication. Indeed, an *active* attacker can act as a MiM and impersonate the verifier, sending $a = \mathbf{0}$, *i.e.*, the all-0 bitstring, to the prover, and expect a response of the form $b \bullet y \oplus \epsilon$, essentially reducing the HB$^+$ protocol to an instance of HB [16]. From here on, if the attacker is passive, in what is now the HB protocol based on a secret y and a blinding factor b, he can learn y from the protocol responses if he can solve an instance of the $LPN_{y,\eta}$ problem. For well-chosen parameters, this takes sub-exponential time at best [4]. Unfortunately, an active attacker does not need to apply this strategy to attack HB$^+$. A more efficient MiM attacks is, for instance, the GRS attack described below.

The GRS attack. For this attack [15], a MiM $\mathcal{A}$ waits to receive each value of a from the verifier, then sends to the prover $\hat{a} := a \oplus \delta$, with δ of $\mathcal{A}$'s choosing. The prover's response is

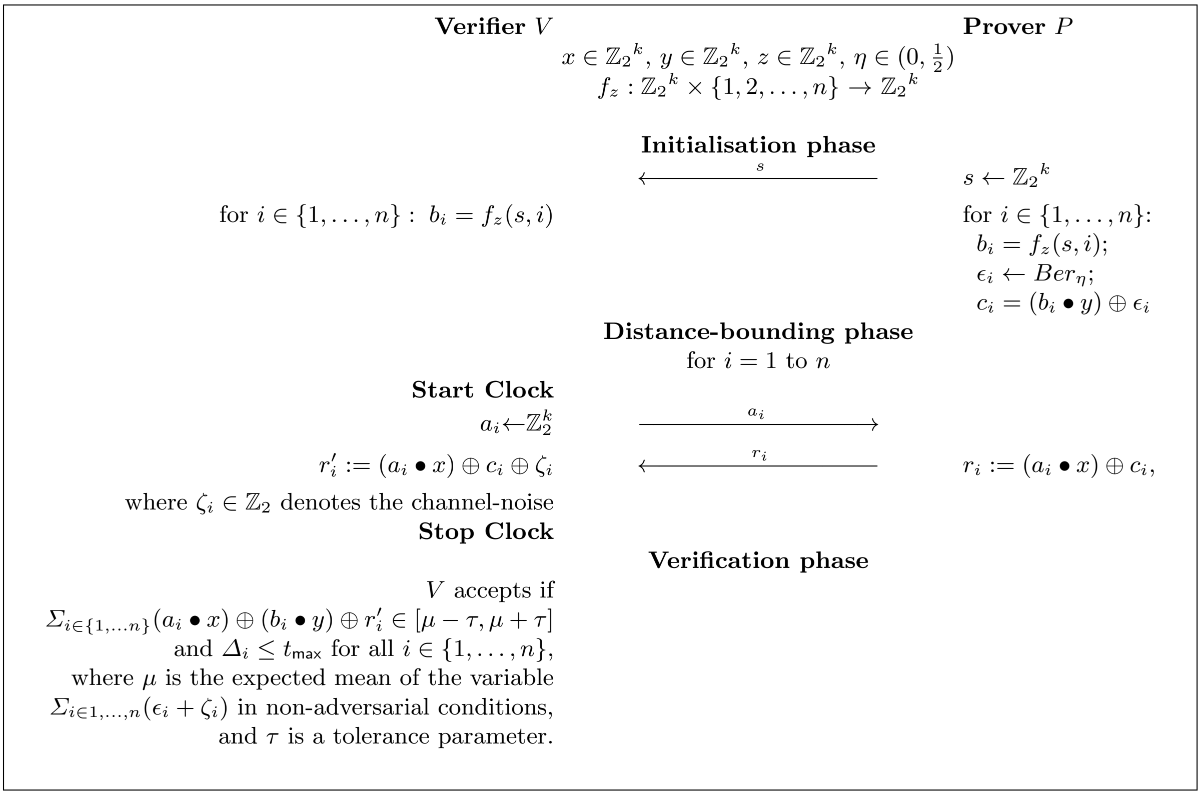

Verifier V
Prover P
$x \in \mathbb{Z}_2{}^k$, $y \in \mathbb{Z}_2{}^k$, $z \in \mathbb{Z}_2{}^k$, $\eta \in (0, \frac{1}{2})$
$f_z : \mathbb{Z}_2{}^k \times \{1, 2, \ldots, n\} \to \mathbb{Z}_2{}^k$
Initialisation phase
s
$s \leftarrow \mathbb{Z}_2{}^k$
for $i \in \{1, \ldots, n\}$: $b_i = f_z(s, i)$
for $i \in \{1, \ldots, n\}$:
$b_i = f_z(s, i)$;
$\epsilon_i \leftarrow Ber_\eta$;
$c_i = (b_i \bullet y) \oplus \epsilon_i$
Distance-bounding phase
for $i = 1$ to n
Start Clock
$a_i \leftarrow \mathbb{Z}_2^k$
a_i
$r_i' := (a_i \bullet x) \oplus c_i \oplus \zeta_i$
r_i
$r_i := (a_i \bullet x) \oplus c_i$,
where $\zeta_i \in \mathbb{Z}_2$ denotes the channel-noise
Stop Clock
Verification phase
V accepts if
$\Sigma_{i \in \{1,\ldots n\}}(a_i \bullet x) \oplus (b_i \bullet y) \oplus r_i' \in [\mu - \tau, \mu + \tau]$
and $\Delta_i \leq t_{\mathsf{max}}$ for all $i \in \{1, \ldots, n\}$,
where μ is the expected mean of the variable
$\Sigma_{i \in 1,\ldots,n}(\epsilon_i + \zeta_i)$ in non-adversarial conditions,
and τ is a tolerance parameter.

Fig. 2: The HB+DB protocol.

not modified by $\mathcal{A}$. The adversary merely waits for the verifier's authentication output. At the end of n rounds, if the verifier has accepted the prover's authentication attempt, then $\mathcal{A}$ learns a part of the secret key (typically, a single bit). In particular, if a specific bit of x is 0, then changing the component of a that is multiplied by that bit makes no difference towards the final result. If the prover authenticates (despite the modified a), then the adversary concludes the corresponding bit of x is 0; otherwise, the reverse statement is assumed to be true. Progressively, by sending linearly independent δ values, $\mathcal{A}$ recovers more bits of the key x. Finally, once the entire key is learned, the adversary can impersonate the prover even without knowing the key y. In particular, by successively sending $b = 0^k$, $\mathcal{A}$ ensures that the authentication process only relies on the knowledge of x.

The HB+DB protocol. Pagnin *et al.* [20,21] proposed a means of preventing MiM attacks such as GRS onto HB$^+$, by enhancing it with a distance-bounding dimension. The resulting protocol, HB+DB, is depicted in Fig. 2.

In HB+DB, the prover and the verifier use three, k-bit long secret keys x, y, z. There are two public noise parameters: the parameter $\eta \in (0, 1/2)$ used in HB$^+$, and the mean $\psi \in [0, 1/2]$ of a Bernoulli distribution modelling channel noise (indicating errors in transmission). Finally, both parties use a function $f_z : \mathbb{Z}_2{}^k \times \{1, 2, \ldots, n\} \to \mathbb{Z}_2{}^k$ keyed with the shared z value, which maps a bitstring of length k to another string of the same length. In analyses in [20,21], this map is assumed to be a pseudorandom function (PRF).

The protocol proceeds in three stages. In the *initialisation phase*, the prover chooses a k-bit string s and, for each $i \in \{1, \ldots, n\}$, computes the following: a k-bit string $b_i := f_z(s, i)$[4]; an "LPN-noise" bit ε_i chosen from a Bernoulli distribution with parameter η; and a bit $c_i = (b_i \bullet y) \oplus \varepsilon_i$. The verifier computes b_i for the received s value, but is oblivious to the chosen LPN noise. In each round of the *distance-bounding phase*, the verifier starts his clock (added from HB$^+$ into HB+DB), then sends a k-bit challenge string a_i. The prover computes and sends a response $r_i := (a_i \bullet x) \oplus c_i$. Note that r_i is one bit long, just like c_i. Since r_i is sent across a noisy channel, the verifier receives a value r_i' adjusted with a noise variable ζ_i, chosen from a Bernoulli distribution with a mean of ψ. Upon receiving $r_i' := r_i \oplus \zeta_i$, the verifier stops the clock and stores the round time Δ_i. During the *verification phase*, the verifier checks that the received responses r_i' were within a certain tolerance from noiseless values $(a_i \bullet x) \oplus (b_i \bullet y)$; the tolerance parameter depends on the two noise factors. Namely, the verifier checks that $\Sigma_{i \in \{1, \ldots n\}} (a_i \bullet x) \oplus (b_i \bullet y) \oplus r_i' \in [\mu - \tau, \mu + \tau]$, where μ is the expected mean of the random variable $\Sigma_{i \in 1, \ldots, n} (\epsilon_i + \zeta_i)$. The verifier also checks that the measured RTTs Δ_i were below $2t_{\mathsf{max}}$, where t_{max} is HB+DB's proximity bound.

3 HB+DB's security shortcomings

The HB+DB protocol [20,21] is essentially a white-box composition of an authentication protocol (*i.e.*, HB$^+$) and a proximity-checking phase. As a consequence, HB+DB can be viewed either as an authentication protocol (as its HB$^+$ precursor), or as a distance-bounding scheme, providing stronger guarantees. As we show in this section, HB+DB guarantees neither of these security notions. We describe flaws at both the authentication and distance-bounding levels, from a provable-security perspective and as practical attacks (of the very threat-type which HB+DB set out to counteract).

3.1 Poor Asymptotic Security in HB+DB

False rejection in "noisy" DB. For DB protocols executed over noisy channels, the *probability* $\mathbb{P}_{\mathsf{Corr}}$ *of correct authentication* of an honest prover placed in the verifier's proximity generally depends on the number of responses that were unperturbed by the channel noise. If too many responses are affected, then an honest prover in the verifier's proximity can be wrongly rejected. We denote the probability of such *false rejection* by $\mathbb{P}_{\mathsf{FR}}$. We clearly have $\mathbb{P}_{\mathsf{FR}}{=}1 - \mathbb{P}_{\mathsf{Corr}}$.

False acceptance in "noisy" DB. In DB executed over noisy channels, the verifier accepts the prover depending on a tolerance parameter. As this acceptance-tolerance parameter becomes more permissive or the noise increases, it becomes more likely for far-away, dishonest provers be falsely authenticated. *I.e.*, the *false acceptance* probability $\mathbb{P}_{\mathsf{FA}}$ increases [11]. This equates to enlarged chances to commit distance fraud, or for a MiM or mafia-fraudster to impersonate the prover.

Tuning parameters in "noisy" DB. It is essential to tune the protocol parameters (*e.g.*, the verifier's acceptance-tolerance, number of rounds, or LPN noise-levels) so that one tightly guarantees low false-rejection and false-acceptance rates. A well-tuned false-acceptance rate entails optimised mafia- and distance-fraud resistance [7].

[4] This is one variant of HB+DB's PRF-instances f_z.

The HB+DB Case. Although Pagnin *et al.* do account for noise when upper-bounding the probability of distance fraud [20,21], they do not concretely propose parameters that would be suitable for the protocol's *secure* deployment, with a reasonable false-rejection probability. Despite having *k-bit challenges*, HB+DB uses *single-bit responses*, which are strongly affected by high noise-levels. What is more, HB+DB's authors requires the presence of LPN-driven noise, in addition to the intrinsic channel noise. As such, the two combined noise-factors negatively impact the correctness, but more seriously, the false-acceptance rate of the protocol.

Indeed, by studying the false-acceptance rate, we ascertained this weakness of the HB+DB protocol. For instance, when instantiated with a 25% LPN-noise (which is standard for HB$^+$) and a generous false-rejection rate of 1%, the *false-acceptance probability* of HB+DBis as high as 2^{-5}, even for a total of 128 time-critical rounds. What is even more worrying is that the false acceptance rate is only a lower bound for the success of both distance-fraud and MiM attacks.

We now formally present the relationship between HB+DB's security and the chosen parameters.

False rejection in HB+DB. The tolerance τ in HB+DB must account for both the *channel noise* –modelled as bits ζ_i following a Bernoulli distribution of parameter ν, and the *LPN noise* – represented as bits ϵ_i as per a Bernoulli distribution of parameter η.

Let S be the random variable described by $\sum_{i=1}^{n}(\epsilon_i \oplus \zeta_i)$. This Bernoulli-distributed[5] variable, of parameter α, is:

$$\alpha := \eta + \nu - 2 \cdot \eta \cdot \nu. \tag{1}$$

The mean of this variable S is $\mu = n\alpha$, in which n is the total number of time-critical rounds.

Thus, the probability $\mathbb{P}_{\mathsf{FR}}^{HB+DB}$ of *false rejection in HB+DB* is the probability that the number of errors lies outside the interval $[\mu - \tau, \mu + \tau]$, namely:

$$\mathbb{P}_{\mathsf{FR}}^{HB+DB} = \sum_{i=\mu-\tau}^{\mu+\tau} \binom{n}{i} \cdot \alpha^i \cdot (1-\alpha)^{n-i}.$$

False-acceptance rates in HB+DB. Let us now look HB+DB's false-acceptance rate. Let r_i^* be the response selected at random by a malicious prover or a malicious MiM for time-critical round i, and denote by r_i the response the verifier expects in that round, based on c_i. Let the random variable s_i^* be described as follows: $s_i^* := 1$ if $r_i^* = r_i$ (the prover was right), and $s_i^* = 0$ if $r_i^* \neq r_i$ (the prover was wrong). Since the responses r_i are pseudorandom, it holds that $\mathbb{P}[s_i^* = 0] = \mathbb{P}[s_i^* = 1] = \frac{1}{2}$. Define $S^* := \Sigma_i s_i^*$.

So, the probability $\mathbb{P}_{\mathsf{FA}}^{HB+DB}$ that this adversary is *falsely accepted in HB+DB*, thus committing DF or MF is:

$$\mathbb{P}_{\mathsf{FA}}^{HB+DB} = \sum_{i=\alpha\cdot n-\tau}^{\alpha\cdot n+\tau} \mathbb{P}[S^* = i] = \sum_{i=\alpha\cdot n-\tau}^{\alpha\cdot n+\tau} \binom{n}{i} \left(\frac{1}{2}\right)^i \left(1-\frac{1}{2}\right)^{n-i}, \tag{2}$$

in which τ is a fixed ratio of n, as computed above.

Tuning HB+DB's parameters. In this study, we do the following:
1) we fix $\mathbb{P}_{\mathsf{FR}}^{HB+DB}$ to the reasonable bound of 1%;
2) we set numbers of rounds, LPN noise and channel noise parameters;

[5] Indeed, in HB+DB a perturbed DB response occurs iff. exactly one of ϵ_i and ζ_i is 1. This happens with probability $\mathbb{P}[\epsilon = 1] \cdot \mathbb{P}[\zeta = 0] + \mathbb{P}[\epsilon = 0] \cdot \mathbb{P}[\zeta = 1] = \eta \cdot (1-\nu) + (1-\eta) \cdot \nu = \eta + \nu - 2 \cdot \eta \cdot \nu$.

3) using these, we **select τ such that $\mathbb{P}_{\mathsf{FR}}^{HB+DB} \approx 1\%$**, and **look at HB+DB's probability of false acceptance**.

The false acceptance rate is a lower bound for the best distance- and mafia-fraud. It represents the success probability of an adversary committing distance- or mafia-fraud/a MiM attack by sending (early) random responses. The latter is the best strategy for successful distance-fraud against HB+DB, according to [20,21]. Since the protocol's challenges are k bits long and the responses are 1-bit long, even attackers have a better strategy in guessing the DB responses rather than the challenges[6].

As far as the noise-parameters are concerned, Pagnin *et al.* [21] specifically require a reasonably high value for η, since *"otherwise the LPN-security is lost"* c.f. [21], page 11. This is inherent to protocols relying on the security of HB$^+$against passive attackers, which resides on LPN-hardness. For instance, [4] shows that, when one bit of the key is guessed, an *LPN* instance with a 32-bit-long key and 5% LPN-noise can be solved in a lower-than-expected time-complexity (in 2^{11}). Typically [19], the LPN noise parameter is set to $\frac{1}{8}$ or $\frac{1}{4}$.

HB+DB's Distance-fraud and Mafia-fraud Asymptotic Resistance. We now report on our study. That is, Fig. 3 shows how the false-acceptance rate given in equation (2) (*i.e.*, when the attacker guesses the responses) varies with noise. More specifically, each of the curves represents how a specific choice of parameters η and ν influences the false-acceptance probability (*i.e.*, DF/MF-resistance), when the number of DB rounds n varies from 32 to 128. In Fig. 3, the values we consider are $\frac{1}{4}$ and $\frac{1}{8}$ for η, and 0.05 and 0.1 for ν. Let us now discuss Fig. 3.

First consider the case of a fixed $\nu = 0.05$. For $\eta \in \{\frac{1}{4}, \frac{1}{8}\}$, both graphs show an (expected) almost-linear descent. However, note that even for high values of n, such as 32 or 64, we still have a very low distance-fraud (and mafia-fraud) resistance. If $\eta = 1/4$, $\nu = 0.05$, and $n = 32$, an attacker has a probability of around 1/2 to succeed in a distance- or mafia-fraud attack by randomly guessing the responses. If $\eta = 1/4$, $\nu = 0.05$, and $n = 64$, the probability of a successful attack becomes 1/8. Even for $n = 128$, the attack still succeeds with almost 2^{-8} probability.

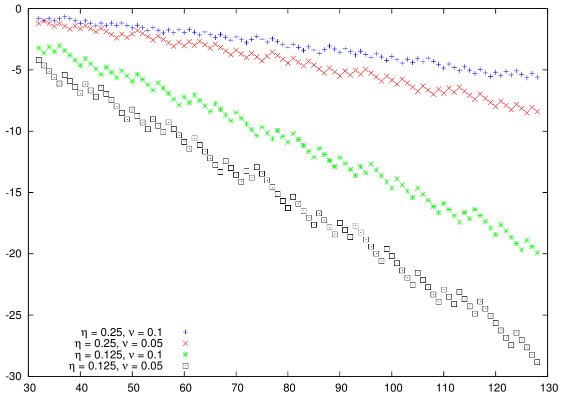
Choosing $\eta = 1/8$ improves HB+DB's false-acceptance rates, since it lowers the tolerance τ. However, the result is still far from ideal, with a resistance to distance- or mafia-fraud (by random response-guessing) of around 5 bits for 32 rounds, 13 bits for 64 rounds, and only 28 bits for 128 rounds. In both cases, therefore, guaranteeing even a low security level (32 bits) requires a lot more rounds. The latter would be exacerbated if the adversary had multiple attempts, which we did not account for in the depicted graphs.

Now consider a fixed $\nu = 0.1$. The first of the two curves takes $\eta = 1/4$, whereas the second takes $\eta = 1/8$. As expected, for equivalent LPN noise levels, these experiments show worse results than for $\nu = 0.05$ (since the total noise has increased). Specifically, with $\eta = \frac{1}{4}$ and $\nu = 0.1$, we observe 2.5 bits of security for 64 rounds, and only around 6 bits for 128 rounds. Similarly, for $\eta = \frac{1}{8}$ and $\nu = 0.1$, the security level for 128 rounds is only of about 19 bits. More importantly, as shown in equation (1), the effect of η and τ on the tolerance rate is symmetric.

We also show how the HB+DB's false-acceptance rate (*i.e.*, DF- and MF-resistance) varies as a function of the LPN-noise parameter η. In Fig. 4, we fix the channel-noise parameter to $\nu = 0.05$, and a generous number $n = 128$ of DB rounds. The curve depicted in Fig. 4

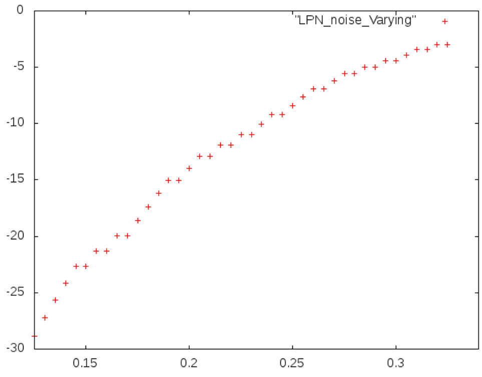
[6] However, in Section 3.2, we do show more powerful/impactful MiMs than just guessers. In that sense, our analysis here best fits the case of distance fraud.

Fig. 3 HB+DB's false-acceptance rates (*i.e.*, success-rates at DF and MF by random response-guessing) for n rounds (from 30 to 130), with $\eta \in \{0.125, 0.25\}$ and $\nu \in \{0.05, 0.1\}$.



shows how poorly the false-acceptance rate scales for increasing values of LPN noise. However, Pagnin *et al.* explicitly require a higher LPN noise to better hide the x, y values; at the same time, the more rounds are used, the more information can potentially leak. In particular, as η approaches 1/3 the adversary has almost a probability of 1 to succeed in the distance-fraud or the mafia-fraud attempt by random response-guessing. Even at $\eta = 0.125$, these resistances are only of about 30 bits of security; however, instead of presenting a curve (indicating a good asymptotic behaviour), the graph shows a nearly linear trend, with about 15 bits of security at $\eta = 0.2$ and 5 bits of security at $\eta = 0.3$.

Fig. 4 HB+DB's false-acceptance rates (*i.e.*, success-rates at DF and MF by random response-guessing) for a fixed $n = 128$ rounds and $\nu = 0.05$, with η varying in the interval $[\frac{1}{8}, \frac{1}{3}]$.



Causes and impact. False-acceptance rates (and so distance- and mafia-fraud resistance) always decrease in the presence of noise. However, HB+DB has a critically-poor robustness to noise. The three main culprits are: (1) the addition of a non-negligible LPN noise; (2) a verification of a *sum* of noisy responses, rather than individual values.

Comparison with other DB protocols. The HB+DB protocol is not the only protocol in the literature to require a correctness threshold of 1%. However, for the same correctness, and for a noise level as high as 0.3, other protocols offer a much higher distance-fraud resistance than HB+DB: *i.e.*, 20 bits of security for about 90 rounds for DB1 (q=3) vs only 6 bits of security for the same number of rounds in HB+DB.This is particularly due to HB+DB's feature of verifying the sum of errors against a tolerance-level.

Removing LPN-noise from HB+DB. Pagnin *et al.* specifically required a high value for η to avoid losing LPN security [21]. However, as we describe in detail in Section 3.4, no such noise is in fact needed for security against passive adversaries. Thus, we also discuss parameter choices for HB+DB in the absence of LPN-noise, for a tolerance rate of at most τ errors. In this case, if the adversary commits DF or MF by randomly guessing the responses, it succeeds if, and only if, the winning session has less than τ errors, yielding a probability of $\sum_{i=0}^{\tau} \mathbb{P}[S^* = i]$.

Like in the "noisy case", in this case we first tune the parameters. To keep a false rejection rate below 1% and taking $n = 92$ and $\nu = 0.05$, we get 10 as the optimal value for τ, *i.e.*, V should then tolerate up to 10 errors. With such parameters, the false-acceptance rate of HB+DB is around 2^{-49}. Under the same conditions ($n = 92$ and $\nu = 0.05$), the DB1 [7] protocol with q=3, yields a significantly lower false-acceptance rate (thus also a DF/MF-resistance) of 2^{-90} compared to 2^{-49}. Moreover, DB1 (with $q = 3$) is computationally more efficient than HB+DB. Thus, even in the absence of noise, the HB+DB protocol compares unfavourably with other distance-bounding protocols, which are additionally more efficient.

Nonetheless, a clear take-away message is that, for the high LPN-noise prescribed by the HB+DB authors, then the protocol's security-levels are unacceptably low (even beyond a computational load yielded by its k-bit long challenges, three k-bit long keys, intense PRF usage, etc.).

3.2 Key-Learning in HB+DB

We now show that the added a proximity-checking countermeasure to HB$^+$ is not sufficient: HB+DB is still vulnerable to other active and practically feasible key-leakage attacks.

MiM attacks in DB: Formalisms & Practice. In DB, the verifier's clock can at most guarantee that relaying (through a MiM attacker) is detected when the prover is far from the verifier. This might not hold when the prover is close to the verifier, as the attacker has more time to act. To increase its time margin, a MiM can *further* exploit side-channel information on the prover's computations [11]. Such intricate MiM attacks are captured by formal DB security models [6].

However, practical experiments by Pagnin *et al.* indicate that the verifier's clock actually detects the MiM attackers who try to modify the challenges. In essence, demodulating the challenge before modifying it and sending it to the prover costs the attacker too much time and leads to detection.

In what follows, we demonstrate that a MiM attacker does not actually *need to demodulate* the challenge to mount a key-recovery attack. As such, adding a proximity-checking phase to authentication protocols such as HB$^+$ does *not* trivially counter active MiM attacks, contrary to the claims in [20,21]. This also highlights one of the dangers of only focusing on particular attacks, rather than providing a provable security guarantee. Whilst worst-case adversaries are often assumed far too strong compared to practical ones, the latter should be limited to one strategy (e.g, demodulation and bit-flipping) as the former are not. Hence, the guarantees provided by a provably-secure protocol are always a safer bet.

Our MiM Attack onto HB+DB. In the following, we describe our concrete MiM attack against HB+DB.

Assumptions and setting. The original HB+DB paper considers NRZ-encoded, ASK modulated challenges: thus, each bit is independently encoded into a high- or low-amplitude signal on the carrier link. We make the following additional assumptions on the adversary: (i) during its attack, $\mathcal{A}$ can "speak louder" than the verifier, *i.e.*, send a signal with higher power, drowning out (part of) V's message, and (ii) $\mathcal{A}$ knows the time interval between 2 challenges, and the bit period. Neither of these is a strong assumption, if the adversary is located between a verifier and the prover. However, condition (i) might only hold in a probabilistic fashion, if different modulation schemes are used. Note also that even if the time interval between challenges is not known *à priori* to the adversary, $\mathcal{A}$ can deduce it after observing one or more sessions.

The attacker's strategy. Instead of reading the challenge as it is transmitted and flipping one of its bits, our adversary will simply *inflict* a particular value (e.g., 1) onto one bit of the challenge received by the prover, for a given round. To do so, $\mathcal{A}$ simply emits a signal stronger than the verifier's signal, at a specific time.

Our active MiM adversary will perform its authentication-attack in two steps. In the first step, *key-recovery*, the honest prover needs to be within the verifier's proximity, but the adversary will be placed between the two honest parties (*e.g.*, close to the legitimate verifier). The second step is *impersonation*: by using the learned key, the attacker impersonates the prover, regardless of the latter's position.

Key-recovery in HB+DB. With $P, \mathcal{A}$, and V positioned as detailed above, the adversary now injects a 1 as the bit at position j in *each* challenge a_i received by the prover for a given session. Recall that $\mathcal{A}$ does so not by *fliping* a bit in the verifier's challenge but by emitting a 1 value "more loudly" at the point corresponding to the j-th bit-period within challenge a_i. Thus, it can escape detection by the proximity-checking countermeasure. As such, the prover receives a modified a_i value, in which the j-th bit is replaced by a 1. Subsequently, the adversary observes the output bit at the end of the authentication attempt. It will eventually conclude that the j-th bit of the secret key x is $x_j = 0$ if, and only if, the authentication is successful. Else, $\mathcal{A}$ sets x_i to 1. Repetition of the attack will eventually allow $\mathcal{A}$ to predict the entire key x.

Let us now see the success probability of such an attack. There are two possibilities for each authentication attempt.

- If the j-th bit of the secret key x is 0, *i.e.*, $x_j = 0$, then injecting a 1 in the challenge at position j does not alter the expected response. Indeed, $0 \cdot a_{i,j}$ is 0 regardless of the value of bit $a_{i,j}$. Hence, if $x_j = 0$, the prover is accepted despite the attack, and the adversary deduces $x_j = 0$ from the result $OutV = 1$. The bit guessed by the adversary is correct with probability $1 - \mathbb{P}_{\mathsf{FR}}$, *i.e.* if the responses of the legitimate prover are accepted (no false rejection occurs)..
- If $x_j = 1$, then the response r_i contains an error in two cases: if $a_{i,j}$ was originally 0, and no LPN or channel noise corrected the error introduced by $\mathcal{A}$, or if $a_{i,j}$ was originally 1, but it was affected by noise. In this case, the probability that a prover answers wrongly is $\frac{1}{2} \cdot \zeta + \frac{1}{2} \cdot (1 - \zeta) = \frac{1}{2}$. This results in the session containing an expected $\frac{n}{2}$ errors, instead of μ, which will likely cause the verifier to refuse the authentication. In this case, the attacker can deduce that $x_j = 1$. The probability for the prover to be rejected in this scenario is the probability that the number of successes of a binomial experiment with n trials and a success probability of $\frac{1}{2}$ does not fall within the interval accepted by the verifier

$[n \cdot \alpha - \tau, n \cdot \alpha + \tau]$. Hence, the probability for the guess of $\mathcal{A}$ to be correct is exactly $1 - \mathbb{P}_{\mathsf{FA}}$.

Since, for random keys, these two scenarii are equiprobable, the probability to recover one bit of x with this attack is $\mathbb{P}_{\text{active}} = \frac{1}{2} \cdot (1 - \mathbb{P}_{\mathsf{FR}}) + \frac{1}{2} \cdot (1 - \mathbb{P}_{\mathsf{FA}}) = 1 - \frac{\mathbb{P}_{\mathsf{FR}} + \mathbb{P}_{\mathsf{FA}}}{2}$, if $\mathbb{P}_{\mathsf{FR}} < \frac{1}{2}$ (otherwise, a rejection by the verifier brings less information as it can occur even for legitimate provers). This is not a limitation, as a protocol wih correctness lower than $\frac{1}{2}$ would be of little practical use. In essence, lowering the false acceptance rate, e.g., by increasing the number of rounds or reducing the LPN noise parameter, inevitably leads to easier key recovery attacks. We further discuss this below.

The feasibility of key-recovery vs. HB+DB's robustness. This attack is feasible in practice and –given its instantaneous bit-changes– bypasses the experimental results of [21,20]. Indeed, the adversary needs not wait to receive, then demodulates the challenge (as for GRS); it simply injects its own modification to the challenge at the right moment.

To see the attack's feasibility/impact, let us see how it varies with the (optimal) security parameters.

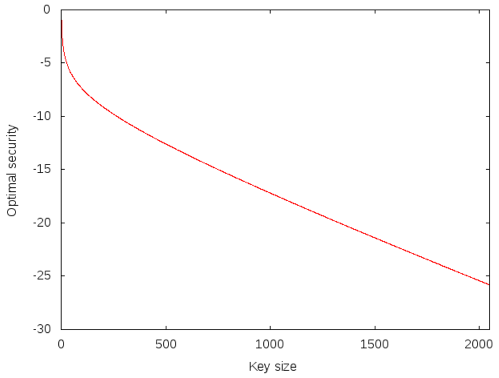
Either Key-Recovery or High False-Rejection Rate. The attacker's probability to recover one bit of the secret key by using this section's strategy is $\mathbb{P}_{\text{active}} = 1 - \frac{\mathbb{P}_{\mathsf{FR}} + \mathbb{P}_{\mathsf{FA}}}{2}$. Since $\mathbb{P}_{\mathsf{FA}}$ and $\mathbb{P}_{\text{active}}$ vary in opposite directions, one can obtain an upper bound on the security of the protocol by looking at their intersection, *i.e.*, the value for which $\mathbb{P}_{\mathsf{FA}} = \mathbb{P}_{\text{active}}$.

Recovering one bit of the key. By solving $\mathbb{P}_{\mathsf{FA}} = \mathbb{P}_{\text{active}}$, we obtain $1 - \frac{\mathbb{P}_{\mathsf{FR}} + \mathbb{P}_{\mathsf{FA}}}{2} = \mathbb{P}_{\mathsf{FA}} \leftrightarrow 2 - (\mathbb{P}_{\mathsf{FR}} + \mathbb{P}_{\mathsf{FA}}) = 2 \cdot \mathbb{P}_{\mathsf{FA}} \equiv 3 \cdot \mathbb{P}_{\mathsf{FA}} = 2 - \mathbb{P}_{\mathsf{FR}} \equiv \mathbb{P}_{\mathsf{FA}} = \frac{2 - \mathbb{P}_{\mathsf{FR}}}{3}$. So, we have that: **regardless of the choice of the other parameters, the adversary is able to either authenticate by sending random responses or recover one bit of the key with a probability p such that $p \geq \frac{2 - \mathbb{P}_{\mathsf{FR}}}{3}$.** If we take $\mathbb{P}_{\mathsf{FR}} = 0.01$, we have $p \geq 0.66$. We insist that this is only a lower bound on the success probability of the best attack, which is independent of the chosen parameters. To lower this probability, one can only increase the false rejection rate, which poses problems w.r.t. the usability/robustness of the protocol.

Recovering the whole key. Let us now see what happens if the adversary has access to k sessions, where k is the size of the key x. We consider that he wins if, among these k sessions, he can either (i) authenticate once or more with random responses, or (ii) recover x. The probability of (i) occurring is $1 - (1 - \mathbb{P}_{\mathsf{FA}})^k$, i.e., 1 minus the chance to fail k times. The probability to recover the key, that is of (ii) happening, is equal to $(\mathbb{P}_{\text{active}})^k = (1 - \frac{\mathbb{P}_{\mathsf{FR}} + \mathbb{P}_{\mathsf{FA}}}{2})^k$. Note that (i) and (ii) increase and respectively decrease with p for positive values of k. Thus, at their intersection lies the best achievable security bound, independently of the choice of parameters. Fig. 5 shows this probability for a wide range of key-sizes, from 1 to 2048 bits. It shows that even for a 2048-bit key, HB+DB cannot achieve more than 26 bits of security while rejecting no more than 1% of legitimate authentication attempts. This renders the protocol hardly useable in comparison of other DB protocols, which achieve significantly higher security levels, even for keys of 128 bits, and have a significantly lower computational overhead.

Quick remedies. Preventing the active attack herein could be done by providing a formal security proof should, since that would guarantee security against *arbitrary* adversarial strategies. Alongside, to prevent transcript malleability attempts, an option is to add a transcript-authentication mechanism at the end of the protocol, *à la* Swiss-Knife [18].

Fig. 5 The base-2 log of a lower bound for the best attack probability on HB+DB for $\mathbb{P}_{\mathsf{FR}} = 0.01$ and k sessions, for each key size k.



3.3 On the Key-based Security in HB+DB

The attack presented in Section 3.2 leads to the recovery of a single key x out of the three (denoted x, y, z) used by HB+DB. In this section we show, however, that recovering x is sufficient to break both the authentication and distance-bounding properties of the protocol.

To see this, assume the existence of an attacker $\mathcal{A}$ that knows/recovers the key x of a prover P. Using the view of $\mathcal{A}$, we now construct an active MiM attacker $\mathcal{B}$ which impersonates an honest, far-away P without being detected by the proximity-checking countermeasure.

In practice, $\mathcal{B}$ would be composed of two distinct entities communicating with each other: one near the far-away prover, and one near the verifier.

The attack proceeds as follows. Given the key x output by $\mathcal{A}$ (and nothing else), an adversary $\mathcal{B}$ starts a session sid with a far-away, honest P, and a separate session sid' with V. Its goal is to make V output 1 at the end of sid'. In the slow phases of both sid and sid', $\mathcal{B}$ just relays s from P to V. This is not detected by the verifier's clock, since it is a not a time critical exchange. In the fast phase, $\mathcal{B}$ first plays out its session with P, sending all a_i equal to 0, thus getting $b_i \cdot y + \epsilon_i$ from P. We can safely assume a noise-cancelling device used by $\mathcal{B}$ on this stretch (or the two devices can just be in each other's very close proximity). Next, as $\mathcal{B}$ receives valid a_i values in sid' (from V), it uses x and the values obtained from session sid, namely $b_i \cdot y + \epsilon_i$ for each i, to construct the expected $(a_i \bullet x) \oplus (b_i \bullet y) \oplus \epsilon_i$ responses. Since this is the responses that V expected, $\mathcal{B}$ succeeds with a probability equalling the correctness of the protocol (with respect to the tolerance threshold and the LPN noise).

Impact. The presence of the three secret keys in this protocol is thus deceptive. On the one hand, it may seem that choosing keys such that their total length is large is sufficient to provide security; on the other hand, it may also seem that leaking (part of) one key is not too serious a threat, as long as the rest of the keys are safe. However, the analysis above shows that the true security of the protocol hinges exclusively on the security of x.

Quick remedies. One option is to prevent transcript malleability, by adding an authentication of the session transcript. Another aspect to consider is: if the key y, which is used in the DB phase, adds no extra MiM security as explained above, then its actual place in the protocol is debatable.

3.4 HB+DB is Not LPN-based

The very high false-acceptance rate described in Section 3.1 strongly depends on the LPN noise. While the authors of HB+DB state that such noise is necessary for the the security of HB+DB against passive adversary (see p.11 of [21]), we show the contrary: this security does not reduce to the hardness of the LPN instance underlying it.

HB$^+$ and LPN. If an HB$^+$ execution [17] were used in the absence of LPN noise (*i.e.*, if η=0), then a passive adversary against the protocol would be faced with a set of linear equations of the form $a_i \bullet x \oplus b_i \bullet y = r_i$, for publicly-known a_i, b_i, and r_i values. As a consequence, a passive adversary observing $poly(n)$ sessions can solve this system and break security. This is why HB$^+$ requires a non-zero LPN noise ϵ, and one can show that an HB$^+$ execution with non-zero noise is as secure against passive adversaries as the hardness of the LPN instance $LPN_{x,\eta}$ used within.

HB+DB and LPN: The Informal Issue. Unlike HB$^+$, even a noiseless instance of HB+DB resists passive attacks. This is because a passive adversary against HB+DB is faced with a set of equations of the form $a_i \bullet x \oplus b_i \bullet y = r_i$, but in which only a_i and r_i are public, whilst the b_is remain *secret*, known only to the honest parties. Thus, b_i randomises the padding to $a_i \bullet x$, which turns an honest execution of HB+DB into a computationally-hard problem for the observing adversary, without it being based on a hard instance of LPN.

HB+DB and LPN: Formalising the issue. Let P and V be two honest parties, and x, y, z be correctly generated in a fixed HB+DB execution. The noise bits ϵ_i are sampled by P following Ber_η. As P is honest and thus uses the PRF-instance f_z correctly, the vectors b_i produced are (indistinguishably close to being) uniformly distributed over $\mathbb{Z}_2^k$, yielding values $(b_i \bullet y)$ that are indistinguishable from uniformly sampled bits. Therefore, the bits c_i that P produces in this way follow a uniform distribution, as each c_i acts as an one-time pad encryption of ϵ_i under the key $b_i \bullet y$. Then, a passive adversary against HB+DB is faced with equations of the form $a_i \bullet x \oplus c_i = r_i$, where a_i and r_i are known, but the uniformly random c_i values are not known. Moreover, all a_i and c_i values change from session to session. Thus, solving the system of equations for x is synonymous to breaking one-time pad security with respect to the plaintext $a_i \bullet x$, the key c_i, and the ciphertext r_i. Moreover, the winning probability in this game is independent of the value η chosen for the distribution of the LPN noise. In other words, the success of a passive attacker against HB+DB does not depend on the security on the underlying $LPN_{x,\eta}$ instance.

Causes and impact. The main reason why HB+DB's security against passive adversaries cannot be reduced to the hardness of LPN, despite claims to the contrary in [20,21], is that the b_i values are only known to the two honest parties, but not to the observing attacker. Incidentally, HB+DB's security against passive adversaries is not lost, relying instead on the pseudorandomness of the function f. But we can argue that the preservation of security is a fortunate accident in the case of HB+DB, and it is dangerous to be mistaken in the assumptions underlying the security of a protocol. This is where formal proofs would help.

4 A way to fix HB+DB

Our observations on the (in)security of the HB+DB protocol raise two natural questions. The first is whether distance-bounding *can* be used to secure HB$^+$ against active adversaries. The second is how to fix the HB+DB protocol, *i.e.*, how to design a similar scheme, but which guarantees provable authentication- and distance-bounding security.

We do not provide a concrete answer to the first of these questions, with Section 3.1 and Section 3.2 just touching upon this matter just for the particular case of HB+DB. It is not trivial to see how to compose a LPN-based passive-secure authentication (*e.g.*, by means of HB$^+$) with proximity-checking, since the necessity for a non-negligible amount of noise for the former creates a window of opportunity for impersonation- and distance-fraud attackers in the latter.

In this section, we answer the second of these questions, and present a protocol called $\mathbb{BLOG}$, which has a lighter computational-load than HB+DB, while offering provably-secure distance-bounding guarantees. In particular, $\mathbb{BLOG}$ is also a secure authentication protocol.

4.1 The $\mathbb{BLOG}$ protocol

For our protocol, $\mathbb{BLOG}$, we rely on our analyses of HB+DB in Section 3 to simplify the latter's structure as follows.

Fig. 6 $\mathbb{BLOG}$: A DB Protocol Issued From HB+DB

Verifier V		**Prover** P
	$x \in \mathbb{Z}_2{}^k$	
	$f_x : \mathbb{Z}_2{}^k \to \mathbb{Z}_2{}^k$	
	Initialisation phase	
	$\xleftarrow{\quad s \quad}$	$s \leftarrow \mathbb{Z}_2{}^k$
$(x_{\mathsf{temp}}\vert\vert b) \leftarrow f_x(s)$		$(x_{\mathsf{temp}}\vert\vert b) \leftarrow f_x(s)$
V halts if $HW(x_{\mathsf{temp}}) = 0$		
	DB phase	
	for $i = 1$ to n	
Start Clock		
$a_i {\leftarrow} \mathbb{Z}_2^k$	$\xrightarrow{\quad a_i \quad}$	
	$\xleftarrow{\quad r_i \quad}$	$r_i := (a_i \bullet x_{\mathsf{temp}}) \oplus b_i$
calculate Δ_i as the		
a_i–r'_i RTT time		
Stop Clock		
	Verification phase	
V accepts if $\forall i \in \{1, \ldots, n\}$:		
$r_i = (a_i \bullet x_{\mathsf{temp}}) \oplus b_i$		
and $\Delta_i \leq t_{\mathsf{max}}$		

Removing the LPN noise. We begin by removing the LPN-noise used in HB+DB's responses. Recall that Section 3.4 shows that no protective layer is added to HB+DB by the LPN-noise and it in fact weakens the security guarantees of the protocol, allowing for an easier false-acceptance, as described in Section 3.1.

Removing the key y. With the LPN-noise parameter set to 0, the response function in HB+DB yields $r_i = a_i \bullet x \oplus b_i \bullet y$. Since for each time-critical round i, the value b_i acts as a one-time-pad to $a_i \bullet x$, we can achieve the same effect by simply drawing b_i at random, and dispensing with the key y. In this way, we save $n \cdot k$ bits of storage, and a total of n dot-product computations (in n time-critical rounds).

Addressing active attacks & removing the key z. A classical DB countermeasure to almost-optimally deter mafia fraud (and thus provide security against all active attacks, including GRS and the one showed above) is to authenticate the entire session transcript at the end of the session [8,18]. However, such messages make the terrorist-fraud resistance proofs problematic, see *e.g.*, [13].

We instead choose to follow the approach of Avoine *et al.*. in their protocol TREAD [1], and ensure that the key which is susceptible to recovery attacks, x,is replaced by an one-time random string x_{temp}. Thus, even if some bits of that value leak, the adversary cannot gain more information by observing other sessions using the same key x_{temp}.

In $\mathbb{BLOG}$, we will no longer compute the inner-product of the challenges a_i with x (as in HB+DB), but rather the inner-product of a_i with the *random, one-time, prover-chosen* x_{temp} value. The key x is now employed exclusively to freshly generate new values of x_{temp}, in a similar manner as the z value used to generate b_i in HB+DB. To this end, HB+DB's key z is removed as well. This new protocol structure also allows us to prove $\mathbb{BLOG}$'s terrorist-fraud resistance in the same way as that of TREAD. Namely, a prover-aided attacker will be able to simply re-use any information learned from the prover in a new session, with the same temporary secret. The same strategy cannot be used effectively by a MiM attacker because the latter has not been given the time-critical responses, which the terrorist will need in order to authenticate.

One-bit responses. As we aim to remain close to the HB+DB design, we preserve its structure with k-bit challenges and 1-bit responses. However, as opposed to HB+DB, which fails to achieve its optimal mafia- and distance-fraud bound of about n bits of security, our protocol $\mathbb{BLOG}$ very nearly reaches that optimal bound, and in a provable way.

The $\mathbb{BLOG}$ ***protocol***. Our $\mathbb{BLOG}$ protocol is depicted in Figure 6. As shown, the prover and verifier now only share one long-term value x only (which will play the part of the key z in HB+DB). During the *initialisation phase*, the prover picks a random k-bit value s, as in the HB+DB protocol, and it sends this value to V. Both compute the output of $f_x(s)$, which is the concatenation $x_{\mathsf{temp}}||b$ of the k-bit long value x_{temp} and the n-bit long value b. If $x_{\mathsf{temp}} = 0$ (as indicated by its Hamming weight $HW(x_{\mathsf{temp}})$), V aborts the execution[7].

The *distance-bounding phase* consists of n time-critical exchanges. For each round, V picks a k-bit value a_i uniformly at random, starts its clock, and sends a_i to P. The prover is expected to reply with $r_i = a_i \bullet x_{\mathsf{temp}} \oplus b_i$ (*i.e.*, the inner-product $a_i \bullet x_{\mathsf{temp}}$ xor-ed with the i-th bit of b); upon receiving this value, V stops its clock and stores the elapsed time Δ_i.

Finally, in the *verification phase*, V checks the correctness of the received r_i, and that $\Delta_i \leq 2t_{\mathsf{max}}$ for each round, and if so, V returns an accepting bit. *I.e.*, the verifier accepts if it holds that $\forall_{i=1}^{n}(\,(r_i = a_i \bullet x_{\mathsf{temp}} \oplus b_i)$ and $(\Delta_i \leq t_{\mathsf{max}})\,)$.

4.2 The security of $\mathbb{BLOG}$

We now outline $\mathbb{BLOG}$'s security properties. The proofs are given in Appendix A.

Theorem 1. *For the* $\mathbb{BLOG}$ *protocol, if the key x is chosen uniformly and independently at random and the challenges a_i are picked independently and uniformly at random by the honest verifier, then the following statements hold:*

[7] Typically, this only happens with a probability of $\frac{1}{2^k}$ for honnest provers.

DF Resistance: $\mathbb{BLOG}$ *is* $q_{\mathsf{V}}\left(\frac{1}{2}\right)^n$*-distant fraud resistant to any adversary opening at most* q_{V} *adversary-verifier sessions.*

MF Resistance: *If in addition f is a secure PRF, then, for any (q_{obs}, q_{P}, q_{V})-mafia fraud adversary $\mathcal{A}$ against $\mathbb{BLOG}$, there exists an adversary $\mathcal{B}$ against the security of f such that:*

$$\mathbb{P}[\mathcal{A}\ wins] \leq (q_{\mathsf{obs}} + q_{\mathsf{P}})^2 \cdot 2^{-k} + \mathsf{Adv}_{\mathcal{B}}^{\mathbb{PRF}} + q_{\mathsf{V}} \cdot \left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^n$$
$$+ (q_{\mathsf{obs}} + q_{\mathsf{P}}) \cdot 2^{-k},$$

TF Resistance: $\mathbb{BLOG}$ *is* SimTF*-resistant.*

4.3 An Evaluation of $\mathbb{BLOG}$

In this section we evaluate our protocol from the point of view of complexity and with respect to channel noise.

Complexity & security. Our $\mathbb{BLOG}$ protocol keeps the strong terrorist-fraud resistance offered by HB+DB, while it adds near-optimal mafia- and distance-fraud security. To our knowledge, $\mathbb{BLOG}$ is the only distance-bounding protocol to exhibit such strong properties *while also guaranteeing provable terrorist-fraud resistance.* If we consider proofs of terrorist-fraud resistance in other models [6], then the DB1 protocol in [7] (for q=3) is the closest to $\mathbb{BLOG}$ security-wise but it is computationally more efficient. If we overlook attacks by programmable PRFs [5], on the basis that they can be bypassed by a well-designed block-cipher, and the lack of a formal proof for the terrorist-fraud resistance property for the Swiss-Knife protocol [18], then the latter is also comparable with $\mathbb{BLOG}$ from the perspective of provable security.

However, note that our protocol does not do as well as it could, specifically because we preserve the k-bit challenge/1-bit response structure of HB+DB. Typically, for a k-bit challenge *and* a k-bit response we could hope for a higher security bound of 2^{-kn}; as it is, the additional complexity caused by the size of the challenges has no apparent benefit.

In terms of complexity, the $\mathbb{BLOG}$ protocol requires the prover to perform k `AND` operations, and k `XOR` operations to compute each of the n time-critical responses, as opposed to just using lookup tables or making one single `XOR` operation per round in most distance-bounding protocols. Since for HB+DB and $\mathbb{BLOG}$, k represents the bit-length of the key, the latter can today be no lower than, say, 80 bits (to prevent trivial brute-force strategies). So, HB+DB and $\mathbb{BLOG}$'s computational complexity cannot be easily lowered, which may mean that their proximity-checks not being as practical.

Channel-noise in $\mathbb{BLOG}$. Our protocol and its security proofs did not treat the case of noisy communications. If we augmented the $\mathbb{BLOG}$ design to be robust to channel-noise, then we would need to change the verification phase as follows: (1) require that responses be verified one by one, but that only a fraction l out of n rounds yield correct responses; (2) require that all responses be within the time-bound.

As such, each bound close to 2^{-n} attained for DF and MiM-resistance would remain the dominant factor in these resistance-bounds, yet each would (provably) change to $\mathsf{Tail}(n, m, 1-p_{noise})$, where $\mathsf{Tail}(n, m, 1-p_{noise})$ is the tail of the binomial distribution denoting the probability of at least m successes occurring over n trials and $1 - p_{noise}$ is the chance of one individual success hinging on a response-bit b not being flipped due to the channel-noise. Indeed, both in the DFKO model [12] and the Boureanu et al. model [6], we could provably show $\mathbb{BLOG}$'s DF/MF-resistance for this noisy case. We adjourn the proofs of DF/MiM/TF resistances in the "noisy" case to an extended version of this paper.

5 Conclusions

Lightweight authentication protocols such HB or HB$^+$ [16,17] were designed specifically for resource-constrained devices, relying on the computationally inexpensive learning parity with noise (LPN) problem. The hardness of LPN makes HB and HB$^+$ provably-secure against *passive impersonation* attempts; yet, neither protocol is secure against *active attacks* [15]. By adding a distance bounding dimension to HB$^+$, Pagnin et al. [20] aimed to achieve active impersonation-security for their HB$^+$-hybrid, called HB+DB.

Apart from losing its lightweight character through extensive use of a pseudorandom function and numerous time-critical computations, HB+DB comes with a number of serious flaws. In this paper, we showed that HB+DB still does not prevent active MiM attacks. What is more, we exhibit that security against these active attacks scales poorly with the robustness of the HB+DB. For instance, even for a very large key-size of 2048 bits, HB+DB can achieve a maximum of 26 bits of security (against our active attack), if HB+DB is to reject no more than 1% of legitimate authentication attempts. We also demonstrated that the added noise factors (as required by Pagnin et al. in [21]) makes HB+DB remarkably prone to distance and mafia-fraud by random response-guessing, even for a very high number of rounds. We have finally proved that –despite the claims of Pagnin et al. in [21]– HB+DB lost HB$^+$'s feature of having the security against passive attackers rely on LPN hardness.

We describe one possible fix of the HB+DB protocol, which is provably secure in the DFKO model [12]. Our proposal, $\mathbb{BLOG}$, has nearly-optimal distance-fraud and mafia-fraud resistance and it is terrorist-fraud resistant. $\mathbb{BLOG}$ is more efficient than HB+DB but not sufficiently lightweight for resource-constrained devices. In *proving* the security of our protocol we do not criticize the experimental approach taken by Pagnin *et al.*, instead we suggest that experimentation must also be supported by a provable security analysis.

A composition between HB$^+$ and distance-bounding without losing the LPN-based passive security (if possible) remains open, as do the full depths of the relations between LPN-based security and distance-bounding security.

References

1. Gildas Avoine, Xavier Bultel, Sébastien Gambs, David Gérault, Pascal Lafourcade, Cristina Onete, and Jean-Marc Robert. A terrorist-fraud resistant and extractor-free anonymous distance-bounding protocol. Accepted at ASIACCS 2017, available at http://onete.net/papers/anondb.pdf, 2017.
2. Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 232–249, 1993.
3. Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on*, 24(3):384–386, May 1978.
4. Sonia Bogos and Serge Vaudenay. Optimization of LPN solving algorithms. Cryptology ePrint Archive, Report 2016/288, 2016. `http://eprint.iacr.org/2016/288`.
5. I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the pseudorandom function assumption in (Secure) distance-bounding protocols. In *Progress in Cryptology – LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 100–120. Springer Verlag, 2012.
6. Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. Practical and Provably Secure Distance-Bounding. *Journal of Computer Security*, 23(2):229–257, 2015.

7. Ioana Boureanu and Serge Vaudenay. Optimal proximity proofs. In *10th International Conference, Inscrypt 2014, Beijing, China,*, pages 170–190. Springer International Publishing, Cham, 2015.
8. Stephen Brands and David Chaum. Distance-bounding protocols (extended abstract). In *Proceedings of EUROCRYPT*, LNCS, pages 344–359. Springer, 1993.
9. Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *Proceedings of TCC*, pages 17–33. Springer, 2005.
10. Julien Bringer Hervé Chabanne and Emmanuelle Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In *Proceedings of SecPerU*, 2006.
11. Jolyon Clulow, Gerhard P Hancke, Markus G Kuhn, and Tyler Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *European Workshop on Security in Ad-hoc and Sensor Networks*. Springer, 2006.
12. Ulrich Dürholz, Marc Fischlin, Michael Kasper, and Cristina Onete. A Formal Approach to Distance Bounding RFID Protocols. In *Proceedings ISC*, LNCS, 2011.
13. Marc Fischlin and Cristina Onete. Terrorism in distance bounding: Modeling terrorist-fraud resistance. In *Proceedings of ACNS*, pages 414–431. Springer, 2013.
14. Henri Gilbert, Matthew Robshaw, and Yannick Seurin. $HB^{\#}$: Increasing the Security and Efficiency of HB^{+}. In *Proceedings of EUROCRYPT*, pages 361–378, 2008.
15. Henri Gilbert, Matthew Robshaw, and Hervé Sibert. Active attack against HB^{+}: a provably secure lightweight authentication protocol. *Electronics Letters*, 41, 2005.
16. Nicholas Hopper and Manuel Blum. Secure human identification protocols. In *Proceedings of ASIACRYPT*, volume 2248 of *LNCS*, pages 52–66. Springer Verlag, 2001.
17. Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In *Proceedings of CRYPTO*, volume 3621 of *LNCS*. Springer Verlag, 2005.
18. C. H. Kim, G. Avoine, F. Koeune, F.X. Standaert, and O. Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *Proceedings ICISC*, volume 5461, 2008.
19. Éric Levieil and Pierre-Alain Fouque. An improved LPN algorithm. In *Proceedings of SCN*. Springer, 2006.
20. Elena Pagnin, A. Yang, Gerhard Hancke, and Aikaterini Mitrokotsa. HB+DB, Mitigating Man-in-the-middle Attacks Against HB+ with Distance Bounding. In *Proceedings of ACM WiSec*, pages 3:1–3:6. ACM, 2015.
21. Elena Pagnin, Anjia Yang, Qiao Hu, Gerhard Hancke, and Aikaterini Mitrokotsa. HB+DB: Distance bounding meets human based authentication. *Future Generation Computer Systems*, pages –, 2016. preprint version.

A Proofs

In this section, we prove Theorem 1, by analysing each of its statements in turn.

Proof ((DF Resistance)). We use the DFKO definition of distance-fraud resistance [12]. Let an arbitrary PPT adversary $\mathcal{A} = P^*$ be a malicious prover in possession of the key x which, according to the definition of DF-resistance in [12], *commits* to the responses for all the time-critical rounds before receiving the challenges a_i and keeps to these committed responses r_i^* to the finish of the DF game.

Then, for each i, $i \in \{1, \ldots, n\}$, it must hold that $r_i^* {=} (a_i \bullet x^*_{\mathsf{temp}}) \oplus b_i^*$, where a_i is sampled uniformly at random in their domain, and x^*_{temp} and b_i^* is produced by $\mathcal{A}$ (potentially by using s and x adaptively) before this sampling.

Equivalently, for each i, $i \in \{1, \ldots, n\}$, it must hold that $\mathcal{A}$ can produce r_i^*, b_i^*, x^*_{temp} apriori to the sampling of a_i such that $r_i^* \oplus b_i^* = a_i \bullet x^*_{\mathsf{temp}} \quad (\mathbf{1})$.

By our hypothesis that the DF game finishes, it follows that the verifier did not halt, hence $HW(x^*_{\mathsf{temp}}) \neq 0$, i.e., at least one bit ${x_{\mathsf{temp}}}^*_j$ of x^*_{temp} is 1 ($j \in \{1, \ldots, k\}$). Since a_i are sampled uniformly at random and independently of x^*_{temp} (and of any adaptively chosen non-zero bit ${x_{\mathsf{temp}}}^*_j$), then the bits generated as $a_i \bullet x^*_{\mathsf{temp}}$ follow the uniform distribution.

So, no matter how $\mathcal{A}$ produces r_i^*, b_i^*, the chances that $(\mathbf{1})$ holds are $\frac{1}{2}$. And such, the probability of any ppt. $\mathcal{A} = P^*$ to pass an adversary-verifier session as per the definition of DF-resistance in [12] is $\frac{1}{2^n}$. This yields the DF-resistance bound.

Proof ((MF Resistance)). We now give the main thrust of the game-transitions in the proof, and the final hybrid argument. Some details linked to the DFKO formalism are skipped in the interest of better comprehension.

$\mathbb{G}_0$ This is the original MF game given in the definition of MF-resistance in [12] cast to the case of $\mathbb{BLOG}$.

$\mathbb{G}_1$ In this game, we assume that the honest prover always generates a new s at each of the $q_{\mathsf{obs}} + q_{\mathsf{P}}$ new sessions in which it engages. This is true up to a collision probability of $\binom{q_{\mathsf{obs}}+q_{\mathsf{P}}}{2}2^{-k}$, which we approximate to $(q_{\mathsf{obs}} + q_{\mathsf{P}})^2 \cdot 2^{-k}$.

$\mathbb{G}_2$ In this game, for each unique value of s, we replace the values $x_{\mathsf{temp}}||b$ by a truly random value. This game-transition relies on the security of the pseudorandom function f used.

$\mathbb{G}_3$ In this game, we allow the adversary to also win by returning the specific session-key $x_{\mathsf{temp}} \in {\mathbb{Z}_2}^k$ which corresponds to a s-value, fixed for a given round. For a fixed s, producing the correct corresponding x_{temp} happens with probability 2^{-k}, yielding a game hop loss of at most $(q_{\mathsf{obs}} + q_{\mathsf{P}}) \cdot 2^{-k}$.

$\mathbb{G}_4$ We will now show that, with negligible advantage added to $\mathcal{A}$, $\mathbb{G}_3$ can be replaced with a new game $\mathbb{G}_4$ where the only change is that an authentication bit equal to 0 to the first untainted adversary-verifier session that the adversary runs with the verifier. The two games are distinguishable if, and only if, $\mathcal{A}$ wins in its first attempt.

Let such a first untainted adversary-verifier session in $\mathbb{G}_3$ have the session-identifier sid. Due to the uniqueness of s in $\mathbb{G}_3$, there are three cases to treat:

(a). there is one unique prover-verifier session with id sid' which shares in $\mathbb{G}_3$ the same s with the adversary-verifier session sid;

(b). there is one unique prover-adversary session with id sid' which shares in $\mathbb{G}_3$ the same s with the adversary-verifier session sid;

(c). there is no prover-adversary session and no prover-verifier session to share in $\mathbb{G}_3$ the same s with the adversary-verifier session sid.

Case (a). This is a case in which $\mathcal{A}$ can only observe the transcript of sid'. If sid' took place *after* sid, then $\mathcal{A}$ had no advantage other than trivial guessing (2^{-kn}). If sid' took place before sid, then some of the challenges of sid' might be repeated in sid (each challenge repeats with a probability of 2^{-k}, independently of any other challenges). This is accounted for by a total advantage per round of $\frac{1}{2^k} + (1 - \frac{1}{2^k}) \times \frac{1}{2} = \frac{2^k+1}{2^{k+1}}$.

Case (b). This case is different from (a) in that $\mathcal{A}$ can now interact in the prover-adversary session sid'. However, the adversary cannot purely relay challenges, and, since it only has one prover-adversary session with a matching s, the adversary's best strategy is to *guess* the correct responses.

Case (c). In this case, $\mathcal{A}$ attempts to win in the first untainted session sid, without having been involved actively or passively in another session with the same s. Its best bet is to guess.

From cases (a)–(c) and by looking at point **(2)** above, it follows that $\mathbb{G}_4$ is just as $\mathbb{G}_3$ (from the adversary's viewpoint) except for a probability-gap of $(\frac{2^k+1}{2^{k+1}})^n$.

$\mathbb{G}_5, \ldots, \mathbb{G}_{3+q_{\mathsf{V}}}$ In each game we make the same argument as in the previous game, but for the 2nd, 3rd, $\ldots$, q_{V}-th untainted session which the adversary runs with the verifier. The arguments follows in a similar way.

So, overall, $\mathbb{P}[\mathbb{G}_0] \leq (q_{\mathsf{obs}} + q_{\mathsf{P}})^2 \cdot 2^{-k} + \mathsf{Adv}_{\mathcal{B}}^{\mathbb{PRF}} + (q_{\mathsf{obs}} + q_{\mathsf{P}}) \cdot 2^{-k} + q_{\mathsf{V}} \cdot (\frac{2^k+1}{2^{k+1}})^n$.

Proof ((TF Resistance)). Here, we mostly sketch the TF proof as it is very similar to the one presented in more details in [1]. We also skip some details linked to the DFKO formalism. We use the SimTF notion from [13], for which we describe the following simulator $\mathcal{S}$: for each adversary-verifier session won by $\mathcal{A}$, $\mathcal{S}$ will open similar sessions and just send the same initial message s that $\mathcal{A}$ sent. This will implicitly replay the same x_{temp} and b values, and the challenges will be answered using the same secret key.

Remember that in the SimTF model, the adversary is not allowed to interact with the prover during time-critical rounds. Hence, the information known by $\mathcal{A}$ at the beginning of the time critical phase is sufficient for it to pass (since we consider winning sessions). With that in mind, we observe that $\mathcal{S}$ plays exactly the same experiment as $\mathcal{A}$: it knows $\mathcal{A}$'s original view, and, since the values a_i will be chosen uniformly and independently at random each time, $\mathcal{S}$is just as likely to receive a challenge a_i that it can answer properly as $\mathcal{A}$ was in the first place. Moreover, the same secret x_{temp} is used as in the session won by $\mathcal{A}$, so for a given a_i, the response of $\mathcal{A}$ is valid for $\mathcal{S}$ too.

Thus, by simply applying the same algorithm as $\mathcal{A}$, $\mathcal{S}$ can win with a probability at least as high as $\mathcal{A}$. Indeed, it can endlessly repeat a session (with different challenges though) that it has a good probability of winning, possibly infering more information after each successful attempt.