IACR Communications in Cryptology
ISSN 3006-5496, Vol. 3, No. 1, 28 pages.

https://doi.org/10.62056/abhey7n4e

Check for updates

Superposition Attacks Against LPN-Based Authentication Protocols

Carlos Cid[1,2], David Elkouss[1] and Manuel Goulão[3]

[1] Okinawa Institute of Science and Technology Graduate University, Japan
[2] Simula UiB, Norway
[3] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract. Quantum security most commonly encompasses only offline passive quantum attacks, where a quantum computer is used by an adversary to solve some computationally hard problem, e.g. factoring or discrete logarithm. However, we are witnessing major efforts for the development and deployment of quantum communication networks, and in this environment, cryptographic protocols may also be implemented in quantum devices. In this new setting, a wider range of *online active* attacks may become possible, for example against targets that may, either deliberately or inadvertently, run a cryptographic scheme in superposition. In this work, we demonstrate that authentication protocols whose security is based on the difficulty of learning linear functions subject to errors may be vulnerable to attacks where adversaries can make queries in superposition — that is, under the so-called "Q2" adversarial model. We do so by describing superposition attacks against a family of symmetric-key authentication protocols based on the LPN problem, a post-quantum cryptography assumption. Our attacks against the HB+ and HB# protocols, both of which have classical proofs of security against active attacks, are based on the Bernstein-Vazirani algorithm, and can efficiently recover the secret key. Despite being conceptually simple, we suggest that our attack techniques might be extended and adapted to also allow for superposition attacks against some modern lattice-based identification and post-quantum signature schemes.

Keywords: LPN Problem · Authentication protocols · Superposition attacks · Quantum cryptanalysis

1 Introduction

Cryptography is among the areas to be most affected by emerging quantum technologies. On the one hand, the advent of large-scale and fault-tolerant quantum computers calls for an extension of the classical adversarial model, where attackers may now apply dedicated quantum algorithms to solve *hard* problems on which current cryptographic schemes are based. For example, Shor's algorithm [Sho94] is known to solve problems like integer factorisation and discrete logarithm in polynomial time (against exponential/subexponential time with the best known classical algorithms). This will make the most widely used public-key constructions based on RSA, Diffie-Hellman, DSA, as well as their Elliptic Curve counterparts, completely insecure against adversaries in possession of a Cryptographically Relevant Quantum Computer (CRQC). This has led to an outburst of activities in the design and analysis of new cryptographic schemes whose security is based on mathematical problems for which there are no known quantum algorithms that can offer a significant

E-mail: carlos.cid@oist.jp (Carlos Cid), david.elkouss@oist.jp (David Elkouss), manuel.goulao@inesc-id.pt (Manuel Goulão)

This work is licensed under a "CC BY 4.0" license.
Received: 2026-01-27 Accepted: 2026-04-23

speed-up compared to the classical counterparts. This field of research is now known as *Post-Quantum Cryptography (PQC)*, and much of the recent work in this area has been motivated by the US National Institute of Standards and Technology (NIST) PQC standardization process, which has selected five public-key post-quantum schemes for standardization [Nat].

On the other hand, the development of quantum networks and quantum communication devices opens for the possibility of new cryptographic constructions that make use of quantum mechanics properties. These new constructions may offer stronger security guarantees or even enable new functionalities when compared to the classical setting. The prime example is Quantum Key Distribution (QKD), which allows parties with access to an authenticated classical channel to establish a shared secret keystream, without the need to rely on computational hardness assumptions (although in practice the required authentication is typically realised using classical cryptography). See [BS16] for more examples.

Developments in quantum technologies may lead to even further applications, where cryptographic schemes are now implemented on quantum devices, and accessible via quantum networks. This in turn brings about a natural extension to the adversary model, as now adversaries can not only use a quantum computer to solve hard problems, but may also be able to *quantumly* interact with these cryptographic schemes. This type of cryptanalysis is often called "quantum chosen message/plaintext/ciphertext", "quantum superposition", or simply "quantum" attacks. Our work fits into this scenario: we consider the security of a family of symmetric-key authentication protocols whose security is based on the hardness of learning linear functions with noise against adversaries who can interact with the scheme via messages in superposition.

Contributions. In this paper we demonstrate that authentication protocols featuring the "commit-challenge-response" structure, and whose security relies on the hardness of learning linear functions subject to error, may in some cases be completely insecure against superposition attacks. Classically we know that introducing just a small number of errors to a linear system of equations, particularly in the setting of a random code, seems to result in a very hard problem for a classical computer to solve. However, we show that when allowed to perform linear operations on arbitrary (quantum) input and some secret, the introduction of error (no matter its rate) does not seem to provide any security.

To illustrate this finding, we consider the "quantum chosen message" attack setting against the HB family of symmetric-key authentication protocols, whose security against classical attacks follows from the hardness of the Learning Parity with Noise (LPN) problem. Our attack technique is based on the Bernstein-Vazirani quantum algorithm, and leads to efficient quantum-classical hybrid attacks that fully recover the secret keys of protocols in the HB family, which have been shown to be *provably* secure against classical active attacks.

As far as we are aware, this is not only the first time superposition attacks are applied against authentication protocols, but it is also the first demonstration of quantum attacks being used against schemes whose classical security holds on the hardness of post-quantum assumptions. In addition, we also briefly discuss how the proposed attack techniques may be extended to potentially apply against a class of public-key authentication schemes, some of which form the basis of recent post-quantum constructions. In our opinion, our results demonstrate the importance of also accounting for quantum attacks when analysing the security of proposed cryptosystems that aim to remain secure in the quantum world.

Organisation. The cryptographic schemes we target in this work to illustrate our superposition attack technique are based on the LPN problem. Thus, in Section 3, we define the LPN problem, discuss the best-known algorithms for solving LPN, and list some variants. In Section 4, we introduce the targeted schemes — the HB family of authentication protocols. We focus on its three most prominent members, namely the

HB, HB+ and HB# protocols, and discuss their security and known classical attacks. In Section 5, we characterise the quantum security model that we consider in this work, together with the concept of superposition attacks. The core of our attack technique, the Bernstein-Vazirani quantum algorithm, is presented in Section 6, where we also describe previous applications of the algorithm to cryptanalysis. In Section 7, key-recovery attacks against the HB (Section 7.1), HB+ (Section 7.2), and HB# (Section 7.3) protocols are established. We close the paper in Section 8 with a discussion of our results and ideas for further work.

2 Preliminaries

Notation. We denote by $\mathbb{Z}_2$ the binary field with two elements. We will interchangeably refer to binary strings in $\{0,1\}^n$ and vectors in the vector space $\mathbb{Z}_2^n$; they will be written in bold font $\mathbf{x}$, and their Hamming weight denoted as $\mathrm{wt}(\mathbf{x})$. We will denote the addition operation in $\mathbb{Z}_2$ and $\mathbb{Z}_2^n$ by $\oplus$. If $\mathbf{a}, \mathbf{x} \in \mathbb{Z}_2^n$, then let $\mathbf{a} \cdot \mathbf{x}$ be their dot (or inner) product. If $\mathbf{x} \in \mathbb{Z}_2^n$ is a vector, we denote by $x_i \in \mathbb{Z}_2$, $1 \leq i \leq n$, the i-th entry of $\mathbf{x}$. We denote binary matrices by capital letters, e.g., $X \in \mathbb{Z}_2^{m \times n}$, and consider X_i to represent either the i-th row or i-th column of X, as it will be explicitly stated. We use the coding theory convention that vectors may be represented by (single) row matrices.
If $\mathcal{X}$ is a probability distribution, then $x \xleftarrow{\$} \mathcal{X}$ means that x is the outcome of a random sample according to $\mathcal{X}$. If $\tau \in [0,1]$, then $\mathrm{Ber}(\tau)$ denotes the Bernoulli distribution with parameter τ; that is, if $x \xleftarrow{\$} \mathrm{Ber}(\tau)$, we have $\Pr[x = 1] = \tau$ and $\Pr[x = 0] = 1 - \tau$. By abuse of notation, if S is a set, we will denote by $s \xleftarrow{\$} S$ the outcome of sampling uniformly at random from S.

Quantum computation. We denote by $\mathbb{C}$ the field of complex numbers and use the standard Dirac notation for quantum states. In our work, we will only make use of pure quantum states $|\psi\rangle$, which describe the state of a physical quantum system and are represented by unit vectors (i.e., $\langle\psi|\psi\rangle = 1$) in a Hilbert space. In particular, we will only use two-level quantum systems, called *qubits* and represented by vectors in $\mathbb{C}^2$. It is customary to denote the canonical basis of this space by $|0\rangle = (1,0)$ and $|1\rangle = (0,1)$, which is usually referred to as the "computational basis". Any qubit may thus be represented as $|\psi\rangle = a\,|0\rangle + b\,|1\rangle = (a,b) \in \mathbb{C}^2$. We denote the "Hadamard basis" vectors by $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. An n-qubit system is represented by a vector in $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 \cong \mathbb{C}^{2^n}$, with the elements of the computational/canonical basis denoted as $|s\rangle \in \mathbb{C}^{2^n}$, where $s \in \mathbb{Z}_2^n$. If $|\psi\rangle \neq |s\rangle$ for all $s \in \mathbb{Z}_2^n$, then the state $|\psi\rangle$ is said to be in *superposition*. The evolution of a quantum system is described by unitary transformations, and at any time the quantum system fully describes the statistical behaviour of a "measurement", as specified by the Born rule. In this work, we will typically represent this evolution using quantum circuits, with unitary operations described as quantum gates. Two gates of particular relevance are the Hadamard gate, which maps between the computational and Hadamard bases, and the CNOT gate, which flips a target qubit conditioned on the value of a control qubit. A comprehensive reference on quantum computation and quantum information can be found in [NC10].

3 The LPN Problem

The *Learning Parity with Noise* problem is one of the most prominent hard learning problems with application in cryptography [BFKL94]. We give below a formal definition of the LPN problem taken from [Pie12].

Definition 1 (decisional/search LPN problem). For $0 < \tau < 1/2$ and $n \in \mathbb{N}$, the

decisional $\mathrm{LPN}_{\tau,n}$ problem is (m, t, ϵ)-hard if, for every distinguisher $\mathcal{D}$ running in time t, we have

$$\left| \Pr_{\mathbf{s},A,\mathbf{e}}[\mathcal{D}(A, A \cdot \mathbf{s} \oplus \mathbf{e}) = 1] - \Pr_{\mathbf{r},A}[\mathcal{D}(A, \mathbf{r}) = 1] \right| \leq \epsilon,$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$, $A \xleftarrow{\$} \mathbb{Z}_2^{m \times n}$, $\mathbf{e} \xleftarrow{\$} (\mathrm{Ber}(\tau))^m$, and $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^m$.
The *search* $\mathrm{LPN}_{\tau,n}$ problem is (m, t, ϵ)-hard if for every adversary $\mathcal{D}$ running in time t,

$$\left| \Pr_{\mathbf{s},A,\mathbf{e}}[\mathcal{D}(A, A \cdot \mathbf{s} \oplus \mathbf{e}) = \mathbf{s}] \right| \leq \epsilon.$$

Informally, hard LPN instances mean that, given m random samples of "noisy" inner products $b_j = \mathbf{a}_j \cdot \mathbf{s} \oplus e_j$ with known (but *not* chosen) $\mathbf{a}_j$, and noise rate τ, there is no efficient adversary that can recover the secret $\mathbf{s} \in \mathbb{Z}_2^n$ (*search*), or even distinguish $\mathbf{b} = (b_1, \ldots, b_m)$ from random (*decision*). These decision and search versions of LPN are polynomially equivalent [BFKL94].

The LPN problem is known to be NP-Hard. The search version can be viewed as the average-case problem of decoding random linear codes of length m: given an LPN instance, we consider $A^\intercal \in \mathbb{Z}_2^{n \times m}$ the code generator matrix and the secret $\mathbf{s} \in \mathbb{Z}_2^n$ as the message to be recovered from the codeword $\mathbf{b} \in \mathbb{Z}_2^m$. The best algorithm for solving LPN is the BKW algorithm [BKW03], which given $2^{O(\frac{n}{\log n})}$ samples can recover the secret $\mathbf{s}$ in time $2^{O(\frac{n}{\log n})}$. A few improvements in the algorithm have been proposed, e.g. [LF06, FMI+06, GJL14], often resulting in more refined practical parameters for LPN-based cryptographic schemes. However, they all still require a sub-exponential number of samples and have sub-exponential time complexity. No quantum algorithms are known that can solve the LPN problem significantly faster than classical ones; see [EKM17, KT17, CT24, SOKB25].

Several variants of the LPN problem have also been proposed, typically with changes in the way the noise is generated. These proposals have been mainly motivated by particular features of the cryptographic systems they are used in. These variants' hardness assumptions are however often less understood and/or less studied. Examples include *exact-noise* LPN (where the Hamming weight of the error vector $\mathbf{e}$ is exactly $\lfloor \tau m \rceil$ [Pie12]), *low-noise* LPN (where $\tau = O(\sqrt{n})$ [Ale03]), RingLPN (where the secret and samples are elements of a quotient of the ring of univariate polynomials over $\mathbb{Z}_2$ [HKL+12]).

We note that the *Learning with Errors (LWE)* problem can be seen as a generalisation of LPN over larger prime fields (and different error distributions). While not as versatile as LWE, the LPN problem has been used to define symmetric-key encryption and authentication protocols, collision-resistant hash functions, pseudo-random generators, public-key encryption, identification and commitment schemes, and zero-knowledge proofs; see [Pie12].

4 The HB authentication protocols

The target of our superposition attacks is the HB-family of symmetric-key authentication protocols, whose security can be directly linked to the hardness of the LPN problem. They were proposed in the early 2000s, and attracted much attention from the cryptographic community due to their simplicity, strong security guarantees, and very lightweight implementation profile. They were at the time believed to be particularly suitable for the authentication of low-end RFID tags, which in most cases could not implement more sophisticated cryptographic protocols. In all schemes described below, $\tau \in (0, \frac{1}{2})$ denotes the parameter of the Bernoulli samplers used in the protocols.

The first member of the HB-family, the *HB authentication protocol*, was proposed by Hopper and Blum at Asiacrypt 2001 [HB01], in the context of "human executable" authentication and identification. The proposed protocol aimed to enable an unassisted

human to repeatedly identify themselves to a Verifier, with provable security against eavesdropping adversaries.

In the HB protocol, a Prover wishes to authenticate itself to a Verifier, with whom it shares a secret binary string $\mathbf{x}$ of length n. It is an interactive, challenge-response protocol, where in each round the Verifier sends a fresh randomly generated challenge $\mathbf{a} \in \mathbb{Z}_2^n$, and the Prover responds with the result of the dot product $\mathbf{a} \cdot \mathbf{x}$, which is flipped with probability τ. One round of the HB protocol is shown in Figure 1. Authentication is successful if after m rounds, the Verifier "check" fails at most τm times.

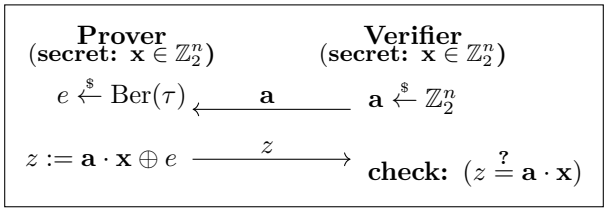

Figure 1: One round of the HB protocol [HB01].

It is straightforward to see that runs of the HB protocol give rise to LPN instances. This in turn leads to a proof of security against *passive* adversaries based on the hardness of the LPN problem. On the other hand, it is also trivial to see that an *active* adversary can efficiently recover the secret $\mathbf{x}$ by interacting with the Prover (Section 4.1).

The *HB+ authentication protocol* was proposed by Juels and Weis at CRYPTO 2005 [JW05], as an extension of the original HB protocol. To protect against active attacks, the parties now share two secrets $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2^n$, and each round of the protocol has one extra message exchange. One round of the HB+ authentication protocol is shown in Figure 2. As before, authentication is successful if after m rounds, the Verifier "check" fails at most τm times.

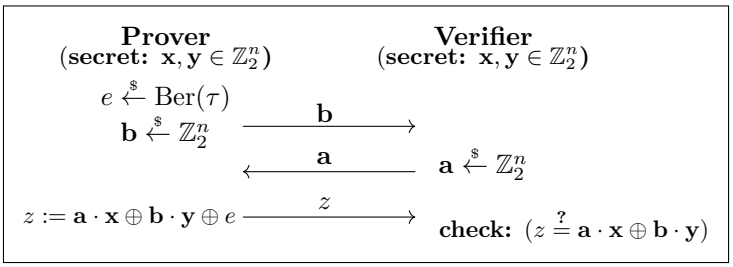

Figure 2: One round of the HB+ protocol [JW05].

The paper proposing the HB+ protocol included a proof of security against active adversaries, also based on a reduction from the LPN problem. The proof was however shortly shown to be flawed, through an ingenious man-in-the-middle (MITM) attack proposed by Gilbert et al. [GRS05] (see Section 4.1). The security proof of HB+ was later simplified in [KSS10], corroborating the claim that HB+ is provably secure against non-MITM active attacks. More precisely, the proof considers active attacks running in two stages: first the adversary may interact polynomially many times with an honest Prover (with concurrent executions allowed), while in the second stage the adversary interacts only with the Verifier [KSS10].

The *HB# authentication protocol*, an extension of HB+, was proposed in 2008 by Gilbert, Robshaw and Seurin at Eurocrypt 2008 [GRS08b]. In their protocol, the shared

secrets are no longer n-bit vectors $\mathbf{x}, \mathbf{y}$, but rather binary matrices X, Y of sizes $(n_X \times m)$ and $(n_Y \times m)$, respectively. The protocol then runs in one single round to decide successful authentication. One may view the HB# protocol as equivalent to m rounds of the HB+ protocol, where each round uses different secrets $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{Z}_2^{n_X} \times \mathbb{Z}_2^{n_Y}$, corresponding to the columns of X, Y. See Figure 3.

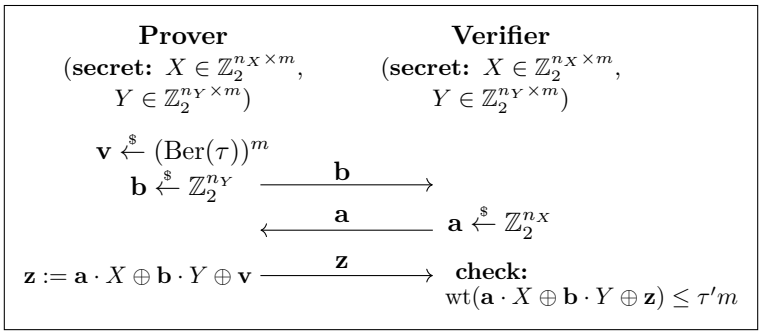

Figure 3: Full run of the HB# protocol. The parameter $\tau' \approx \tau$ is selected based on desired false negative/positive rates.

The HB# protocol comes in two versions. In RANDOM-HB# the shared secrets are two randomly generated matrices. A more memory-efficient version, called simply HB#, was also proposed, where X, Y are Toeplitz matrices. The modifications introduced in the HB# protocol lead to a more robust proof of security against active attacks, including MITM. Indeed, the authors of HB# state that their protocol "remains provably resistant to attacks in the model of Juels and Weis, and at the same time is provably resistant to a broader class of active attacks that includes the attack of [GRS05]". However, Ouafi et al. [OOV08] presented a general man-in-the-middle attack against both versions of the protocol, with practical query complexity for some parameter sets; see Section 4.1.

We do not provide specific parameters for the above protocols (i.e., the concrete values of $n, n_X, n_Y, m, \tau, \tau'$) to achieve particular security levels. These are typically derived from estimates for the concrete complexity of the BKW algorithm, which have often been revised since the protocols were originally published; see [LF06, GRS08b, BTV16] for some examples.

4.1 Classical attacks against the HB protocols

The original HB protocol is trivially broken by an active adversary playing the Verifier: the attacker repeatedly queries the Prover with n linearly independent vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ to derive the values of the n inner products $(\mathbf{v}_1 \cdot \mathbf{x}), (\mathbf{v}_2 \cdot \mathbf{x}), \ldots, (\mathbf{v}_n \cdot \mathbf{x})$ by majority rule. The secret $\mathbf{x}$ can then be recovered by solving a linear system of equations. To recover the value of each $\mathbf{v}_i \cdot \mathbf{x}$, one needs around $(1 - 2\tau)^{-2}$ queries. This means that, for the typical values used for τ in the HB protocol, an active key-recovery attack has data and time complexities of $O(n)$ and $O(n^3)$, respectively.

The HB+ protocol claimed to be provably secure against active attacks, but as noted in [GRS05], the scenarios considered in the security proof in [JW05] did not include the adversary affecting a legitimate run of the protocol. And indeed, the protocol can be efficiently broken by an MITM adversary, as described in [GRS05]: to recover the secret $\mathbf{x}$, the adversary modifies the Verifier challenge $\mathbf{a}$ as $\mathbf{a}' = \mathbf{a} \oplus \mathbf{d}$, using a fixed $\mathbf{d} \in \mathbb{Z}_2^n$ for the m rounds of a run of the protocol, to derive $\mathbf{d} \cdot \mathbf{x}$ based on the outcome of the authentication attempt. For example, sequentially using the canonical basis vectors $\mathbf{d}_i$ allows the adversary to recover each individual bit of $\mathbf{x}$, requiring n runs of the protocol.

Then $\mathbf{x}$ can be used either to impersonate the Prover (by setting $\mathbf{b} = 0$ in authentication attempts), or to recover $\mathbf{y}$ by interacting with the Verifier with n linearly independent $\mathbf{b}$ values. The overall attack remains very efficient: flipping bits in each of the m rounds of n legitimate runs of the protocol to recover $\mathbf{x}$, then interacting with the Verifier with total nm messages to recover $\mathbf{y}$.

The security of the HB# protocol was formally proven in [GRS08b] against a broad class of man-in-the-middle attacks; it was however only conjectured for the general case. And indeed, in [OOV08] a general man-in-the-middle attack against both versions of the protocol was presented. In the attack, the adversary is given the ability to modify all messages. It first recovers the weight of the error vector, which can then be used to recover the error vector itself, leading to full key-recovery. The attack recovers the shared secrets in 2^{25} or 2^{20} authentication rounds for HB#, and 2^{34} or 2^{28} for RANDOM-HB#, depending on the parameter set. See [OOV08] for full details.

In spite of the MITM attacks by [GRS05] and [OOV08] against HB+ and HB#, respectively, we note that the proofs of security for both protocols against classical active non-MITM adversaries remain sound. In particular, the results in those two works do not invalidate the claims of provable security of HB+ and HB# against active attacks in which the adversary may first interact polynomially many times with an honest Prover, and then interact *only* with the Verifier in attempting to defeat the protocols' security. We therefore consider the protocols as legitimate targets to illustrate how our superposition attack technique may defeat provably secure authentication schemes based on the hardness of learning linear functions with errors.

4.2 Other protocols in the HB-family

In this work we chose to focus on the most prominent members of the HB-family — the HB, HB+ and HB# protocols — but for the sake of completeness, we note that several other authentication protocols based on HB have also been proposed. However, most offered weaker security guarantees against classical active attacks. For example, HB++ [BCD06], HB-MP [MP07] and HB* [DK07] were all shown not to meet their original security claims [GRS08a]. Trusted-HB, proposed by Bringer and Chabanne [BC08], added a second stage to the HB+ protocol, to include a universal hash function based MAC to authenticate the protocol run transcripts. However, it was later also shown to be insecure in [FS09], albeit with a high-complexity attack. We do not discuss these protocols in this paper, but note that our quantum attack against HB+ (Section 7) can also be applied to Trusted-HB to recover one of the secret keys, which in turn would allow unlimited impersonation.

5 Quantum security model

The classical security of cryptographic protocols and schemes considers that the participating parties are PPT and all information and communication are classical. However, there is now a need to account for the development of quantum technologies, which may for example give an adversary a larger power ($\mathsf{BPP} \subseteq \mathsf{BQP}$) to solve certain problems that are thought to be classically intractable, for example factoring or computing discrete logarithm. The way to model this new setting is subtle. One may consider the scenario where an adversary can make use of a quantum computer locally to solve a problem that it could not solve in a classical setting. That is, an adversary is allowed to perform quantum computation, but all communication is still classical. This is the adversarial model considered in *post-quantum cryptography*.

As a further natural extension, cryptographic security may need to take into account the possibility that even classical protocols could be run in a quantum network environment,

where participating parties might perform an algorithm (e.g., as a subroutine) in a quantum computer (or might be tricked into doing so), or that an adversary could observe the output of a function before it being measured by an honest party. These legitimate scenarios make the notion of security that considers quantum adversaries, but only classical cryptographic operations unsatisfactory. Moreover, yet another natural generalisation that encompasses all possible quantum scenarios is to allow an adversary not only to run quantum algorithms, but also for the communication to be quantum as well. This model has been extensively studied, and is the standard when considering *quantum security*.

One consequence of this security model is that cryptographic primitives may now operate on data in superposition, giving an adversary the power to infer information through these means. Indeed, [KM10, KM12] (using Simon's algorithm [Sim94]) were the first to show that, by allowing the interaction with a cryptographic primitive to be quantum, constructions that were otherwise secure may become insecure.

A push to formalise quantum security has ensued since. In general, one considers that an adversary has oracle access to a cryptographic function f, to which it is allowed to perform queries in superposition as

$$\mathcal{O}_f : |x\rangle |y\rangle \mapsto |x\rangle |y + f(x)\rangle . \tag{1}$$

Attacks using these types of strategies became known as "superposition attacks" [DFNS14] (regarding secret sharing and multi-party computation), or "quantum chosen message attacks" [BZ13a] (considering message authentication codes), or simply "quantum-security" [Zha12] (for pseudo-random functions). In cryptanalysis research, this model is often referred to as the "Q2 setting" [KLLNP16].

Moreover, [BZ13b] defined "quantum chosen message security" for signature schemes (the adversary receives a superposition of signatures) and "quantum chosen ciphertext security" for encryption schemes (the adversary receives a superposition of decryptions). Their definitions follow the typical IND-CPA and IND-CCA (classical) concepts, but consider different settings for the usual two phases. In the "learning phase", quantum queries are allowed to a given oracle; yet, in the final "challenge phase" only classical messages are allowed, introducing the two new notions of IND-qCPA and IND-qCCA.

In [GHS16], the classical equivalence reductions that relate semantic security and indistinguishability [GM84] are analysed in the quantum setting. In particular, the stronger concept (stronger than IND-qCPA) of "quantum semantic security under chosen plaintext attacks" (qSEM-qCPA), which allows a quantum challenge phase (for semantic security), is introduced for encryption schemes. This definition (qSEM-qCPA) was proved to be equivalent to the concept of qIND-qCPA where now the challenge phase allows quantum messages regarding indistinguishability.

Finally, in [ABF+16], a more general approach to quantum protocols for encryption is taken. Instead of considering the quantum security of classical protocols, symmetric and public-key cryptography schemes are allowed to be fully quantum (i.e., quantum plaintext and ciphertext messages, but classical keys). Again, the classical relation of semantic security and indistinguishability [GM84] is studied, showing that their equivalence holds in the quantum setting.

Current work. We consider the setting where we have quantum access to an oracle evaluating the cryptographic primitive in question, such that we can make queries in superposition, i.e., we work in the Q2 setting. Since we are dealing with symmetric-key authentication (interactive) protocols, the messages exchanged by the parties are used to mask the shared secret, such that a Verifier can assess whether a Prover has knowledge of a shared secret key, without leaking information about the key in the exchanged messages. Since these messages do not represent any plaintext (as it would be in an encryption scheme), we shall refer to the setting where an attacker can maliciously choose these messages as quantum states as "quantum chosen message attack". However, this setting is

analogous to the setting of quantum chosen plaintext attack, qCPA, and thus we may also refer to our security setting as qCPA.
We remark that all attacks presented in our work achieve full key-recovery. As a consequence, our work shows that none of the targeted primitives fulfills quantum chosen message attack security, and therefore do not fulfill any stronger level of security either. In fact, since we can recover the classical key in a learning phase using quantum queries, then in any — even classical — challenge phase we can break the security of the protocols.

6 Bernstein-Vazirani algorithm

The Bernstein-Vazirani algorithm was proposed in 1997 in the seminal work [BV97]. It was designed to show that there is an (oracle) separation between the BPP and BQP complexity classes. For that, they introduce a new oracle problem that requires an algorithm that takes a polynomial (linear) number of queries to the oracle in the size of the problem in a classical computer, but for which there exists a quantum algorithm that requires only one query to the oracle. On top of this, they construct a new problem by encapsulating various recursive levels of the previous problem, such that a quantum (BQP) algorithm is able to solve the problem using a polynomial number of queries to the oracle, while a probabilistic (BPP) classical algorithm provably requires a super-polynomial number of queries to the oracle. We remark that, in our work, we only make use of the first, non-recursive, version of the algorithm, which we describe below.

Problem statement. One is given an oracle that implements a function $f_{\mathbf{s}} : \mathbb{Z}_2^n \to \mathbb{Z}_2$ parameterised by a secret $\mathbf{s} \in \mathbb{Z}_2^n$, which is promised to evaluate to

$$f_{\mathbf{s}}(\mathbf{x}) = \mathbf{x} \cdot \mathbf{s} = \bigoplus_{i=1}^{n} x_i s_i, \tag{2}$$

i.e., the dot product (in $\mathbb{Z}_2$) of $\mathbf{x}$ and $\mathbf{s}$. The goal is to find $\mathbf{s}$.

Classical solution. To solve this problem classically, one needs to construct a linear system of n independent relations, otherwise the problem is underdetermined. The simplest solution is to query the oracle on inputs $\mathbf{x}^{(1)} = (1, 0, 0, \ldots, 0), \ldots, \mathbf{x}^{(n)} = (0, 0, 0, \ldots, 1)$, and recover each bit of $\mathbf{s}$ as the output of the oracle,

$$s_i = f_{\mathbf{s}}\left(\mathbf{x}^{(i)}\right). \tag{3}$$

Thus, one needs to query the oracle n times to recover the full secret $\mathbf{s} \in \mathbb{Z}_2^n$.

Quantum solution. Interestingly, there exists a quantum algorithm, proposed by Bernstein and Vazirani, that recovers $\mathbf{s}$ with a single quantum query to an oracle implementing $f_{\mathbf{s}}$. We describe their algorithm below.

First, assume we are given an oracle, illustrated in Figure 4, that implements the function $f_{\mathbf{s}}$ as

$$\mathcal{O}_{f_{\mathbf{s}}} : |\mathbf{x}\rangle |y\rangle \mapsto |\mathbf{x}\rangle |y \oplus f_{\mathbf{s}}(\mathbf{x})\rangle . \tag{4}$$

We consider the standard phase kickback procedure, and take $|y\rangle = |-\rangle$, making the oracle into a phase oracle as

$$\mathcal{O}_{f_{\mathbf{s}}} : |\mathbf{x}\rangle |-\rangle \mapsto (-1)^{f_{\mathbf{s}}(\mathbf{x})} |\mathbf{x}\rangle |-\rangle . \tag{5}$$

We will often just ignore the (separable) second register, which remains unchanged as $|-\rangle$, and simply have

$$\mathcal{O}_{f_{\mathbf{s}}} : |\mathbf{x}\rangle \mapsto (-1)^{f_{\mathbf{s}}(\mathbf{x})} |\mathbf{x}\rangle . \tag{6}$$

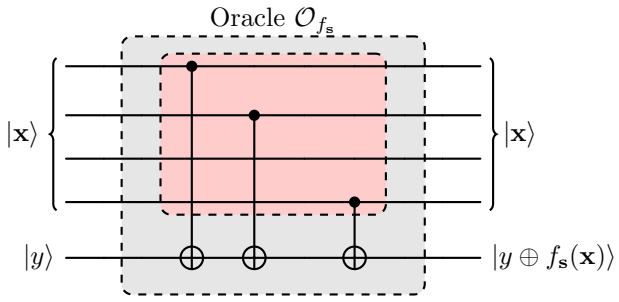

Figure 4: Quantum circuit for the oracle $\mathcal{O}_{f_s}$ (in gray) for the evaluation of the function $f_{\mathbf{s}}(\mathbf{x}) = \mathbf{x} \cdot \mathbf{s}$, for $\mathbf{s} = (1, 1, 0, 1)$ (secret, in red).

Algorithm 1 Bernstein-Vazirani algorithm

Require: register $|\varphi\rangle \in \mathbb{C}^{2^n}$; oracle $\mathcal{O}_{f_s}$
Ensure: $|\varphi\rangle = |\mathbf{s}\rangle$
1: $|\varphi\rangle \leftarrow |0\rangle^{\otimes n}$
2: $|\varphi\rangle \leftarrow H^{\otimes n} |\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x} \in \mathbb{Z}_2^n} |\mathbf{x}\rangle$
3: $|\varphi\rangle \leftarrow \mathcal{O}_{f_s} |\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x} \in \mathbb{Z}_2^n} (-1)^{\mathbf{x} \cdot \mathbf{s}} |\mathbf{x}\rangle$
4: $|\varphi\rangle \leftarrow H^{\otimes n} |\varphi\rangle = |\mathbf{s}\rangle$
5: **return** $|\varphi\rangle$

The Bernstein-Vazirani algorithm works by applying the oracle $\mathcal{O}_{f_s}$ to the uniform superposition of the input domain $\mathbf{x} \in \mathbb{Z}_2^n$, and applying the Hadamard transform to create constructive interference such that its outcome is the state $|\mathbf{s}\rangle$. It can then be measured in the computational basis to learn $\mathbf{s} \in \mathbb{Z}_2^n$.

The Hadamard transform, denoted $H^{\otimes n} : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, applies the Hadamard gate ($H : \mathbb{C}^2 \to \mathbb{C}^2$; $H |x\rangle = (1/\sqrt{2}) \sum_{y \in \mathbb{Z}_2} (-1)^{xy} |y\rangle$) to each of the n qubits individually,

$$H^{\otimes n} : |\mathbf{x}\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y} \in \mathbb{Z}_2^n} (-1)^{\mathbf{x} \cdot \mathbf{y}} |\mathbf{y}\rangle . \tag{7}$$

The Bernstein-Vazirani algorithm works as described in Algorithm 1, when given $\mathcal{O}_{f_s}$ as a black-box implemented as in Equation (6). In Figure 5, we provide a schematic illustration of the Bernstein-Vazirani algorithm in the circuit model of quantum computation.

Example. Suppose that a function $f_{\mathbf{s}} : \{0,1\}^4 \to \{0,1\}$ hides a secret $\mathbf{s} = (1, 1, 0, 1)$ by computing $f_{\mathbf{s}}(\mathbf{x}) = \mathbf{x} \cdot \mathbf{s}$. Classically, to recover $\mathbf{s}$, an Adversary must query $f_{\mathbf{s}}$ at least 4 times: for instance $f_{\mathbf{s}}((1,0,0,0)) = (1,0,0,0) \cdot (1,1,0,1) = \mathbf{s}_0$, $f_{\mathbf{s}}((0,1,0,0)) = (0,1,0,0) \cdot (1,1,0,1) = \mathbf{s}_1$, ... When the function $f_{\mathbf{s}}$ is implemented in a quantum computer, it computes a unitary transformation as in Figure 4 (note the implementation of the internal

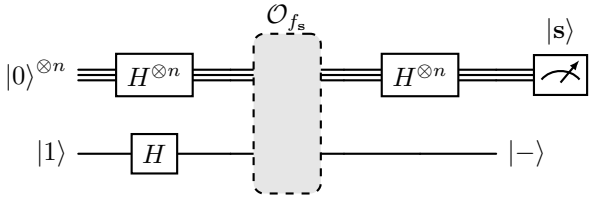

Figure 5: Circuit for the Bernstein-Vazirani algorithm for oracle $\mathcal{O}_{f_s}$ (in gray).

product by multiplying $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_3$ with $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_3$ inside the red box). For this it will have an input register $|\mathbf{x}\rangle$ and an output register $|y\rangle$. Then, an adversary may set $\mathbf{x} = \frac{1}{4}\sum_{\mathbf{i}\in\{0,1\}^4} |\mathbf{i}\rangle$ and $\mathbf{y} = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, such that it is effectively computing $(-1)^{f_{\mathbf{s}}(\mathbf{x})} |\mathbf{x}\rangle |-\rangle$. This is a single query to the function $f_{\mathbf{s}}$, and the adversary can immediately recover $\mathbf{s}$ by applying the Hadamard transform $H^{\otimes 4}$ to the register $|\mathbf{x}\rangle$ and measuring the resulting state $|\mathbf{y}\rangle = H^{\otimes 4} |\mathbf{x}\rangle$. This works because

$$\begin{aligned}
|\mathbf{y}\rangle &= H^{\otimes 4} |\mathbf{x}\rangle \\
&= H^{\otimes 4} \left(\frac{1}{4} \sum_{\mathbf{i}\in\{0,1\}^4} (-1)^{f_{\mathbf{s}}(\mathbf{i})} |\mathbf{i}\rangle \right) \\
&= \frac{1}{16} \sum_{\mathbf{i}\in\{0,1\}^4} (-1)^{\mathbf{i}\cdot\mathbf{s}} \sum_{\mathbf{j}\in\{0,1\}^4} (-1)^{\mathbf{i}\cdot\mathbf{j}} |\mathbf{j}\rangle \\
&= \frac{1}{16} \sum_{\mathbf{i}\in\{0,1\}^4} \sum_{\mathbf{j}\in\{0,1\}^4} (-1)^{\mathbf{i}\cdot(\mathbf{s}\oplus\mathbf{j})} |\mathbf{j}\rangle .
\end{aligned}$$

When $\mathbf{j} = \mathbf{s}$ (so that $(\mathbf{s} \oplus \mathbf{j}) = 0$), the exponent vanishes and the sum over $\mathbf{i}$ evaluates to 16. It follows that the amplitude of $|\mathbf{s}\rangle$ is 1, and hence $|\mathbf{y}\rangle = |\mathbf{s}\rangle$.

6.1 Applications to cryptanalysis

Prior work has used the Bernstein-Vazirani algorithm in cryptanalysis. Interestingly, all the works that target concrete cryptosystems apply the algorithm in the cryptanalysis of symmetric-key primitives, mostly targeting either Feistel or Even-Mansour constructions. We list these below. In contrast, in our work we apply the algorithm for the cryptanalysis of interactive symmetric-key authentication *protocols*, which have been proven secure against classical active attacks.

- [XY19] (extending the original work of [KM10, KM12] where Simon's algorithm is used) applies the Bernstein-Vazirani algorithm as a subroutine to exploit linear structures to distinguish 3-round Feistel constructions and obtain a partial-key recovery attack on Even-Mansour constructions. In addition, an application to differential cryptanalyses is proposed based on this algorithm. The algorithm is then extended in [XY20] to obtain a related-key attack against block ciphers that have a particular linear structure (namely cryptosystems based on the Even-Mansour construction).
- [ZY21] proposes a key-recovery attack on Feistel constructions, by combining the Bernstein-Vazirani algorithm with Grover's algorithm. The authors analyse the attack with respect to different rounds of Feistel structures, motivated by ciphers built on iterating many rounds of this primitive, such as DES.
- [BLNS21] describes a generic "quantum linearisation attack", with applications to multiple proposals of MAC algorithms. They show how to forge valid tags using different quantum algorithms, including the Bernstein-Vazirani algorithm. Their attack exploits functions of the form $G(x_1, x_2, \ldots, x_n) = g(x_1) \oplus g(x_2) \oplus \cdots \oplus g(x_n) \oplus C$, i.e., affine functions with independent constant terms. The authors show that this independent constant term, usually an initialization vector or nonce, which provides protection against forgeries by classical adversaries, does not do so in the qCPA setting.

In a different direction, [GKZ19] discusses "quantum versions" of the LPN and LWE problems. In that work, LPN samples $(\mathbf{a},\ \mathbf{a}\cdot\mathbf{s} \oplus e)$ are assumed to be quantum and

in superposition for different pairs $(\mathbf{a}, e)$, $\sum_{(\mathbf{a},e)} |\mathbf{a}\rangle |\mathbf{a} \cdot \mathbf{s} \oplus e\rangle$. The authors model the error (or "noise") as a deterministic function $e = e(\mathbf{a}, r)$ of the LPN function input $\mathbf{a}$ and randomness r. In this more general case, a different error e is associated to each $\mathbf{a}$, and the output may then be seen as a superposition of multiple independent LPN samples. It is shown in [GKZ19] how to use the Bernstein-Vazirani algorithm to exploit the linear relation between $\mathbf{a}$ and $\mathbf{s}$, and recover $\mathbf{s}$. We note that this result does *not* directly connect to any cryptographic construction based on LPN (or LWE), since the quantum versions of these assumptions cannot (in the general case) be used by classical computers. Moreover, as the authors note in [GKZ19], generating the LPN samples would also require exponential resources. In contrast, our work considers a different setting (the Q2 model) where a classical cryptographic protocol, based on the traditional LPN assumption, is implemented on a quantum computer, likely to operate on classical/basis states, yet also capable of processing queries in superposition.

Peripheral to the field of cryptography, [CSS15] studies the learning of Boolean functions by analysing the problem of learning parity functions, $f_{\mathbf{a}}(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$. In particular, they show that this problem is tractable for both classical and quantum computers (the latter via the Bernstein-Vazirani algorithm), but perturbing the system with noise gives rise to the LPN problem. Thereafter, [CSS15] shows that in the quantum setting the presence of (constant) noise in the output of $f_a(x)$, regardless of its type (classification or depolarising noise) and rate, does not make the problem intractable, and the computational complexity remains polynomial. While the learning scenario analysed in [CSS15] differs from our focus on cryptographic constructions, which explicitly samples the noise for the LPN samples as a component for security using a secret-dependent affine function ($f_s(x) = \mathbf{s} \cdot \mathbf{x} \oplus e$), the underlying phenomenon is the same. It shows that the (constant and independent) noise plays substantially different roles in classical and quantum computations of linear functions.

7 Quantum cryptanalysis of the HB authentication protocols

In this section, we present our superposition attack technique against interactive authentication protocols whose security is based on the hardness of learning linear functions with noise. We do this by showing how an active attacker, who is able to interact quantumly with the Prover, by sending queries in superposition, and then classically with the Verifier, can recover the shared secrets in the HB[1], HB+ and HB# authentication protocols, with very low data and time complexities.

In previous attacks, such as the seminal [KM12], Simon's algorithm is used to target constructions like Even-Mansour with keys k_1, k_2, permutation P, and encrypting message m, defined as $\mathsf{Enc}_{k_1,k_2}(m) = k_2 \oplus P(k_1 \oplus m)$. They work by exploiting a hidden shift, by crafting a function $f_{k_1}(m) = \mathsf{Enc}_{k_1,k_2}(m) \oplus P(m)$, and checking that $f_{k_1}(m) = f_{k_1}(m \oplus k_1)$, and hence recover k_1 using Simon's algorithm. Other attacks, e.g., on MACs, work by finding similar hidden-shift relations [KLLNP16, BLNS21]. In contrast, our attacks employ the Bernstein-Vazirani algorithm, less common in the Q2 model. Below, we will leverage linear relations (in our case, the dot product) perturbed with errors between an adversary-controlled input and the secret key. We will show that these constructions, while achieving provable security in the classical setting,[2] are completely insecure in the Q2 model.

[1] There is only one secret in the HB protocol, thus no interaction with the Verifier is needed for key-recovery.

[2] We note that the HB+/HB# security proofs rely on rewinding, which is a technique that does not easily transfer to the quantum setting. Thus it is unclear whether one may claim that the protocols are provably secure in the "Q1 setting", although no offline attacks against the protocols are known.

Algorithm 2 Key-recovery (quantum) attack on the HB protocol

Require: oracle $\mathcal{O}_P$; quantum register $|\varphi\rangle \in \mathbb{C}^{2^n}$; classical register $\mathbf{r} \in \mathbb{Z}_2^n$
Ensure: $\mathbf{r} = \mathbf{x}$
1: $|\varphi\rangle \leftarrow |0\rangle^{\otimes n}$
2: $|\varphi\rangle \leftarrow H^{\otimes n}|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{a}\in\mathbb{Z}_2^n} |\mathbf{a}\rangle$
3: $|\varphi\rangle \leftarrow \mathcal{O}_P|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}+e} |\mathbf{a}\rangle$
4: $|\varphi\rangle \leftarrow H^{\otimes n}|\varphi\rangle = |\mathbf{x}\rangle$
5: $\mathbf{r} \leftarrow \textbf{measure}(|\varphi\rangle)$
6: **return r**

7.1 HB protocol quantum cryptanalysis: key-recovery in one quantum query

As discussed in Section 4.1, the original HB protocol is not secure even against classical active adversaries, with key recovery expected to require $n \cdot (1-2\tau)^{-2}$ queries. However, to illustrate the main technique used in our work, we describe below how a quantum adversary can use the Bernstein-Vazirani algorithm to recover the secret key $\mathbf{x} \in \mathbb{Z}_2^n$ used in the HB authentication protocol with one single query in superposition. In the subsequent subsections, we apply the same technique to break the active security of different HB variants that have been shown to be provably secure against active classical adversaries.

Recall that the HB protocol [HB01] is an interactive symmetric-key authentication protocol between a Prover P and a Verifier V who share a secret $\mathbf{x} \in \mathbb{Z}_2^n$. It is a challenge-response authentication protocol, parameterised by $n, m \in \mathbb{N}$ and $\tau \in (0, 1/2)$. On a run of the protocol, the Verifier sequentially challenges the Prover with m freshly-generated binary strings $\mathbf{a}^{(i)} \in \mathbb{Z}_2^n$, to which the Prover responds with $z^{(i)} \leftarrow f_{\mathbf{x}}(\mathbf{a}^{(i)})$, where $f_{\mathbf{x}}(\mathbf{a}) = \mathbf{a}\cdot\mathbf{x} \oplus e \in \mathbb{Z}_2$ and $e \xleftarrow{\$} \text{Ber}(\tau)$. The Verifier checks if $z^{(i)} \stackrel{?}{=} \mathbf{a}^{(i)} \cdot \mathbf{x}$ for each round i, and accepts if at most τm rounds fail.

We work in the quantum chosen message scenario and assume that an attacker has quantum access to the protocol that executes with a Prover P implemented as an oracle $\mathcal{O}_P$ for the function $f_{\mathbf{x}}(\mathbf{a})$, as

$$\mathcal{O}_P : |\mathbf{a}\rangle \mapsto (-1)^{f_{\mathbf{x}}(\mathbf{a})} |\mathbf{a}\rangle , \tag{8}$$

where $f_{\mathbf{x}}(\mathbf{a}) = \mathbf{a}\cdot\mathbf{x} \oplus e$ and $e \xleftarrow{\$} \text{Ber}(\tau)$. The function f is probabilistic, however the adversary does not have control or access to the error e; therefore we may consider that this random value is independently sampled before each call to the oracle. This setting is similar to the implementation of a random function by a quantum oracle described in [BLNS21], where each oracle query depends on the previous random choice of an IV. Then, by applying the same procedure as in the Bernstein-Vazirani algorithm (Algorithm 1) to this oracle, we can recover the secret key $\mathbf{x}$ as described in Algorithm 2.

Correctness. In Step 3 of Algorithm 2, the Prover P is run via the oracle $\mathcal{O}_P$. Moreover, Step 4 follows from

$$\begin{aligned} \left|\varphi^{(4)}\right\rangle &= \frac{1}{\sqrt{2^n}} H^{\otimes n} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}\oplus e} |\mathbf{a}\rangle \right) \\ &= (-1)^e \frac{1}{\sqrt{2^n}} H^{\otimes n} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}} |\mathbf{a}\rangle \right) \\ &= (-1)^e |\mathbf{x}\rangle = |\mathbf{x}\rangle . \end{aligned} \tag{9}$$

That is, since $(-1)^e$ is a global phase, the resulting physical state is $|\mathbf{x}\rangle$, and measuring the qubits in the computational basis allows the full recovery of the secret key $\mathbf{x} \in \mathbb{Z}_2^n$.

The key insight of our attack is the following: whereas introducing an affine error term may provide security in the classical setting, by making the problem non-linear (finding an error term of Hamming weight approximately τ), it does *not* provide any security in the quantum chosen message security model. The error e, sampled once as the oracle $\mathcal{O}_P$ is called, is translated into a global phase, which allows the recovery of $\mathbf{x}$ in the same manner as with the Bernstein-Vazirani algorithm. The same phenomenon will be used in the quantum attacks described below against other members of the HB family of protocols; those have however been shown to be provably secure against active adversaries.

We note that this error model in the *classical LPN-based protocols*, which in the Q2 model are quantumly evaluated, is substantially different from the error model of a pure quantum LPN problem, as presented in [GKZ19]. In the former, the Bernoulli-sampled error e is independent of $\mathbf{a}$, even when evaluating the function in superposition. This is a key requirement of the provable security characteristic of LPN-based cryptographic protocols, as the hardness of decoding a random code assumes that the error is independent of the message. Indeed, for a *classical* function $f_{\mathbf{x}}(\mathbf{a}) = \mathbf{a} \cdot \mathbf{x} \oplus e$ to be quantumly evaluated on an input in superposition, the function is only executed once for that single input, and hence the error e is also only sampled once. In contrast, in the latter, the *quantum LPN problem* consists of a superposition of LPN samples $\sum_{(\mathbf{a}_i, e_i)} |\mathbf{a}_i\rangle |\mathbf{a}_i \cdot \mathbf{x} \oplus e_i\rangle$. Moreover, in this scenario, generating LPN samples requires exponential resources. Indeed, constructing a superposition with a distinct independent error for each element of the superposition would require a unitary encoding of all 2^n possible e_i values for each $\mathbf{a}_i$, which in turn requires exponential resources to implement (e.g., a lookup table). [3] The security of this quantum computational assumption is an interesting problem, already analysed in [GKZ19]. However, it is not related to our study of *classical* cryptographic primitives, in this case the HB family of authentication protocols.

Another noteworthy remark is that, while our quantum attack against the HB protocol is a straightforward application of the Bernstein-Vazirani algorithm, it can offer a much higher data complexity advantage compared to the former. The original Bernstein-Vazirani algorithm allows one to recover a secret vector $\mathbf{s}$ with one quantum query against n classical queries. In our case, we have one quantum query against $n/(1-2\tau)^2$ classical queries for the quantum and classical active attacks against the HB protocol, respectively.

7.2 HB+ quantum cryptanalysis: key-recovery in one quantum query and nm classical queries

The HB+ protocol [JW05] is a more interesting target, as it has been shown to be provably secure against active attacks (Section 4). The protocol introduces a modification to the original HB protocol by enforcing both parties to input randomness into the running of the protocol. A round of the protocol starts with the Prover P sending a randomly generated binary string $\mathbf{b} \in \mathbb{Z}_2^n$, before being challenged by the Verifier on $\mathbf{a} \in \mathbb{Z}_2^n$. Like the HB protocol, HB+ is parameterised by $n, m \in \mathbb{N}$ and $\tau \in (0, 1/2)$. The parties now share two secret keys $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$, each corresponding to an LPN sample for each of the input randomness ($\mathbf{a}$ and $\mathbf{b}$) of each party. The challenge-response mechanism executes again in m repetitions, similarly to the plain HB protocol. Now however both randomness from the Prover and the Verifier must be included in the response of P as $z^{(i)} = \mathbf{a}^{(i)} \cdot \mathbf{x} \oplus \mathbf{b}^{(i)} \cdot \mathbf{y} \oplus e^{(i)}$ where $e^{(i)} \leftarrow \mathrm{Ber}(\tau)$. The Verifier V checks $z^{(i)} \stackrel{?}{=} \mathbf{a}^{(i)} \cdot \mathbf{x}^{(i)} \oplus \mathbf{b}^{(i)} \cdot \mathbf{y}^{(i)}$, for $1 \leq i \leq m$, and accepts if at most τm iterations fail.

[3] An efficient alternative would be to use a pseudo-random function to compute $e_i = \mathrm{PRF}_k(\mathbf{a}_i)$ (for another secret key k), but this would break independence of the error and forfeit provable security.

We describe below an active attack against the HB+ protocol, by a quantum adversary in the quantum chosen message attack model. The attack works in two stages to recover the secrets, $\mathbf{x}$ and $\mathbf{y}$, one at a time.

- The first stage, to recover $\mathbf{x}$, consists of a quantum chosen message attack that requires exactly one call to an oracle implementing the Prover in the HB+ protocol.
- The second stage, to recover $\mathbf{y}$, consists of a classical chosen message attack (given that $\mathbf{x}$ is known) that requires $n\,m$ classical queries to a Verifier in the HB+ protocol.

Stage 1: recovery of x. The same technique used in Section 7.1 is employed, based on the Bernstein-Vazirani algorithm and requiring one interaction with an honest Prover. This time the Prover P is assumed to implement a different (but similar) oracle $\mathcal{O}_P : |\mathbf{a}\rangle \to (-1)^{f_{\mathbf{x},\mathbf{b}}(\mathbf{a})} |\mathbf{a}\rangle$, where $f_{\mathbf{x},\mathbf{b}}(\mathbf{a}) = \mathbf{a}\cdot\mathbf{x} \oplus \mathbf{b}\cdot\mathbf{y} \oplus e$. Here $\mathbf{b}$ is the string sent by P to V at the start of the interaction, which together with $\mathbf{x}$, will be used to parameterise the oracle function. Since this part of the attack is, in essence, the same as Algorithm 2, to avoid repetition, we present it in Appendix A as Algorithm 6. It describes how an adversary may interact with an honest Prover P to recover the secret $\mathbf{x} \in \mathbb{Z}_2^n$ in one quantum query to oracle $\mathcal{O}_P$.

Stage 2: recovery of y. Having recovered $\mathbf{x}$, the attacker can *classically* interact with V to recover the second secret $\mathbf{y}$. This stage works similarly to the second part of the MITM attack against HB+ (as alluded in [GRS05]). Essentially, the honest Verifier V is used as an oracle that responds whether messages of the form $z = \mathbf{a}\cdot\mathbf{x} \oplus \mathbf{b}\cdot\mathbf{y} \oplus e$ equals $z' = \mathbf{a}\cdot\mathbf{x} \oplus \mathbf{b}\cdot\mathbf{y}$ for more than $\tau\,m$ rounds of the protocol (which will be the case when authentication succeeds).

The attack requires n runs of the protocol, and is performed as follows: for each run j, let $\mathbf{b}_j$ be the j-th vector of the canonical basis of $\mathbb{Z}_2^n$. The attacker starts all m rounds of the protocol run j with the same $\mathbf{b}_j$. Then, in each round i, on receiving a fresh challenge $\mathbf{a}^{(i)}$ from the Verifier, since the attacker knows $\mathbf{x}$, it can compute $z^{(i)} = \mathbf{a}^{(i)}\cdot\mathbf{x}$, which is sent back to V at the end of the round. Authentication will be decided by checking whether

$$z^{(i)} \oplus (\mathbf{a}^{(i)}\cdot\mathbf{x} \oplus \mathbf{b}_j\cdot\mathbf{y}) = \mathbf{b}_j\cdot\mathbf{y} \stackrel{?}{=} 1$$

at most τm times. Since we use the same $\mathbf{b}_j$ in all m rounds, successful authentication means that $\mathbf{b}_j\cdot\mathbf{y} = 0$; and failure that $\mathbf{b}_j\cdot\mathbf{y} = 1$. Thus, the adversary can recover one bit of $\mathbf{y}$ at a time.

Full recovery of the secret $\mathbf{y} \in \mathbb{Z}_2^n$ requires $n\,m$ queries to an honest Verifier V (which is linear for a constant number of rounds m). The attack is summarised in Algorithm 3. To model the sequential three-step interaction with $\mathcal{O}_V$, we denote the step where the randomness $\mathbf{b}$ is sent from $P \to V$ as $\mathbf{b} \to \mathcal{O}_V^{(P\to V)}$, the step where challenge $\mathbf{a}$ is sent from $V \to P$ as $\mathbf{a} \leftarrow \mathcal{O}_V^{(V\to P)}$, and the step where the response z is sent from $P \to V$ as $z \to \mathcal{O}_V^{(P\to V)}$.

Correctness. Since by the soundness of the HB+ protocol, $\mathcal{O}_V$ accepts if $z = \mathbf{a}\cdot\mathbf{x} + \mathbf{b}\cdot\mathbf{y}$ for at least $(1-\tau)m$ rounds, then, by sending $z = \mathbf{a}\cdot\mathbf{x}$, $\mathcal{O}_V$ accepts if $\mathbf{b}\cdot\mathbf{y} = 0$ for at least $(1-\tau)m$ rounds. By setting $\mathbf{b}_i = 1$ for some position i and zero for all other positions (for all m repetitions), $\mathcal{O}_V$ accepts if the $\mathbf{y}_i = 0$. Hence, for each round j of Algorithm 3, the adversary recovers $\mathbf{y}_i$, and in n rounds recovers $\mathbf{y}$.

7.3 HB# protocol quantum cryptanalysis: full key-recovery in m quantum queries and $\leq (2m+1)n_Y$ classical queries

The HB# protocol is a variant of HB+ that has been proven secure against active attacks in the classical setting (including some types of MITM attacks) [GRS08b]. The protocol

Algorithm 3 Classical attack to recover secret $\mathbf{y}$ in the HB+ protocol

Require: partial key $\mathbf{x} \in \mathbb{Z}_2^n$; oracle $\mathcal{O}_V$
Ensure: $\mathbf{r} = \mathbf{y}$

$\mathbf{r} \leftarrow (1, \ldots, 1) \in \mathbb{Z}_2^n$
for $j = 1$ to n **do**
  $\mathbf{b} \leftarrow (0, \ldots, 0) \in \mathbb{Z}_2^n$
  $b_j \leftarrow 1$
  for $i = 1$ to m **do** ▷ Execute the protocol with V
    $\mathbf{b} \rightarrow \mathcal{O}_V^{(P \rightarrow V)}$
    $\mathbf{a} \leftarrow \mathcal{O}_V^{(V \rightarrow P)}$
    $\mathbf{a} \cdot \mathbf{x} \rightarrow \mathcal{O}_V^{(P \rightarrow V)}$
  end for ▷ End of protocol execution with V
  if $\mathcal{O}_V$ accepts **then**
    $r_j \leftarrow 0$
  end if
end for
return $\mathbf{r}$

executes in only one interaction, which may however be viewed as m queries executed simultaneously. This works by generalising the shared secrets $(X, Y) \in \mathbb{Z}_2^{n_X \times m} \times \mathbb{Z}_2^{n_Y \times m}$ to be matrices. The overall procedure is then the same, but with matrix multiplication that emulates m executions of HB+ for the same challenge vectors $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_2^{n_X} \times \mathbb{Z}_2^{n_Y}$, with an independent random vector $\mathbf{v} \xleftarrow{\$} (\mathrm{Ber}_\tau)^m$. The message sent by the Prover to the Verifier in the authentication attempt is the vector $\mathbf{z} = \mathbf{a} \cdot X \oplus \mathbf{b} \cdot Y \oplus \mathbf{v} \in \mathbb{Z}_2^m$.

Our attack runs again in two stages. In stage 1 (the quantum attack), recovery of X can be achieved using a similar procedure as before. However it now requires interacting with an honest Prover m times, as the secret X is recovered column-by-column. The second stage, for recovery of Y, requires classical interaction with the Prover and an honest Verifier, with at most $(2m + 1)n_Y$ queries.

Stage 1: recovery of X. We first describe a decomposition of the quantum function evaluation oracle for the function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ by explicitly breaking down the output register $|\mathbf{c}\rangle$ "qubit-by-qubit", with $\mathbf{a} \in \mathbb{Z}_2^n$ and $\mathbf{c} \in \mathbb{Z}_2^m$, as

$$\begin{aligned} &\mathcal{O}_f : |\mathbf{a}\rangle |\mathbf{c}\rangle \mapsto |\mathbf{a}\rangle |\mathbf{c} \oplus f(\mathbf{a})\rangle \\ \Leftrightarrow\ &\mathcal{O}_f : |\mathbf{a}\rangle |c_1\rangle |c_2\rangle \ldots |c_m\rangle \mapsto |\mathbf{a}\rangle |c_1 \oplus f(\mathbf{a})_1\rangle |c_2 \oplus f(\mathbf{a})_2\rangle \ldots |c_m \oplus f(\mathbf{a})_m\rangle . \end{aligned} \tag{10}$$

One may then construct a phase oracle as in Equation (6) for just one equation (i.e., one output qubit) by letting the input qubit $|c_i\rangle = |-\rangle$. Notwithstanding, for all other output qubits $|c_j\rangle$ $(j \neq i)$, the output $|c_j \oplus f(\mathbf{a})_j\rangle$ would still depend on the input $|\mathbf{a}\rangle$, i.e., they would be entangled with the input register $|\mathbf{a}\rangle$. This would prevent the constructive interference that results from the second Hadamard transform, and that allows the recovery of the secret (as in Step 4 of Algorithm 1).

However, note that by setting the input qubit $|c_j\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ for $j \neq i$, the resulting state for each qubit after summing $f(\mathbf{a})_j$ is always $|+\rangle$, independently of the value of $f(\mathbf{a})_j$. This results from the fact that adding $f(\mathbf{a})_j = 1$ simply permutes $|0\rangle$ and $|1\rangle$ without introducing any phase, i.e., $\frac{1}{\sqrt{2}}(|0 \oplus f(\mathbf{a})_j\rangle + |1 \oplus f(\mathbf{a})_j\rangle) = |+\rangle$ for any $f(\mathbf{a})_j \in \mathbb{Z}_2$. So, we may now define a phase oracle $\mathcal{O}_f^{(i)}$ that yields only the phase corresponding to the i-th outcome of $f(\mathbf{a})$ as

$$\mathcal{O}_f^{(i)} : \quad |\mathbf{a}\rangle |+\rangle^{\otimes i-1} |-\rangle |+\rangle^{\otimes m-i} \quad \mapsto \quad (-1)^{f(\mathbf{a})_i} |\mathbf{a}\rangle |+\rangle^{\otimes i-1} |-\rangle |+\rangle^{\otimes m-i} , \tag{11}$$

Algorithm 4 Quantum attack to recover secret X in the HB# protocol

Require: oracle $\mathcal{O}_P = \{\mathcal{O}_P^{(i)}\}_i$; quantum register $|\varphi\rangle \in \mathbb{C}^{2^{n_X}}$; classical reg. $R \in \mathbb{Z}_2^{n_X \times m}$
Ensure: $R = X$
1: **for** $i = 1$ to m **do**
2: $\quad |\varphi\rangle \leftarrow |0\rangle^{\otimes n_X}$
3: $\quad |\varphi\rangle \leftarrow H^{\otimes n_X} |\varphi\rangle = \frac{1}{\sqrt{2^{n_X}}} \sum_{\mathbf{a}\in\mathbb{Z}_2^{n_X}} |\mathbf{a}\rangle$
4: $\quad |\varphi\rangle \leftarrow \mathcal{O}_P^{(i)} |\varphi\rangle = \frac{1}{\sqrt{2^{n_X}}} \sum_{\mathbf{a}\in\mathbb{Z}_2^{n_X}} (-1)^{\mathbf{a}\cdot X_i \oplus \mathbf{b}\cdot Y_i \oplus v_i} |\mathbf{a}\rangle$
5: $\quad |\varphi\rangle \leftarrow H^{\otimes n_X} |\varphi\rangle = |X_i\rangle$
6: $\quad R_i \leftarrow \mathbf{measure}(|\varphi\rangle)$
7: **end for**
8: **return** R

which again ignoring the unchanged (separable) second output registers, yields

$$\mathcal{O}_f^{(i)} : |\mathbf{a}\rangle \mapsto (-1)^{f(\mathbf{a})_i} |\mathbf{a}\rangle . \tag{12}$$

An active quantum adversary, playing the role of a Verifier, can interact with an honest Prover P, which is modelled by an oracle $\mathcal{O}_P = \mathcal{O}_{f_{X,Y,\mathbf{b}}}$, with $f_{X,Y,\mathbf{b}}(\mathbf{a}) = \mathbf{a}\cdot X \oplus \mathbf{b}\cdot Y \oplus \mathbf{v}$. Thus, one may define a family of oracles given a quantum chosen message implementation of P, $\{\mathcal{O}_P^{(i)}\}_i$, as

$$\mathcal{O}_P^{(i)} = \mathcal{O}_{f_{X,Y,\mathbf{b}(\cdot)_i}}, \tag{13}$$

with $f_{X,Y,\mathbf{b}}(\mathbf{a})_i$ defined as

$$f_{X,Y,\mathbf{b}}(\mathbf{a})_i = (\mathbf{a}\cdot X \oplus \mathbf{b}\cdot Y \oplus \mathbf{v})_i = \mathbf{a}\cdot X_i \oplus \mathbf{b}\cdot Y_i \oplus v_i, \tag{14}$$

where X_i, Y_i, v_i are the i-th columns of matrices X, Y and the row vector $\mathbf{v}$, respectively. Then, one may perform the Bernstein-Vazirani algorithm to recover $X_i \in Z_2^{n_X}$ by interacting with oracle $\mathcal{O}_P^{(i)}$, as described in Algorithm 4.

Correctness. Note that in Step 4 of Algorithm 4, the Prover P is run via the oracle $\mathcal{O}_P$, but this time the phase oracle from the quantum chosen message only targets the i-th entry of the secret, via the $\mathcal{O}_P^{(i)}$ construction of Equation (11). This effectively creates a phase that only concerns the equation $z = \mathbf{a}\cdot X_i \oplus \mathbf{b}\cdot Y_i \oplus v_i$ (relative to columns i for $X, Y, \mathbf{v}$ as in Equation 14). This is analogous to one iteration of the HB+ protocol. In Step 5 the outcome of the computation is $|X_i\rangle$ because

$$\begin{aligned}
\left|\varphi^{(5)}\right\rangle &= \frac{1}{\sqrt{2^{n_X}}} H^{\otimes n_X} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^{n_X}} (-1)^{\mathbf{a}\cdot X_i \oplus \mathbf{b}\cdot Y_i \oplus v_i} |\mathbf{a}\rangle \right) \\
&= (-1)^{\mathbf{b}\cdot Y_i \oplus v_i} \frac{1}{\sqrt{2^{n_X}}} H^{\otimes n_X} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^{n_X}} (-1)^{\mathbf{a}\cdot X_i} |\mathbf{a}\rangle \right) \\
&= (-1)^{\mathbf{b}\cdot Y_i \oplus v_i} |X_i\rangle \\
&= |X_i\rangle .
\end{aligned} \tag{15}$$

The bit value $\mathbf{b}\cdot Y_i + v_i$ is independent of the secret X and input $\mathbf{a}$, i.e., it is a global phase, and therefore is not observed when measuring the state in Step 6.

In summary, the quantum stage of the attack for recovery of secret X is similar to the attack against HB+ in Algorithm 6 repeated m times, one for each of the columns of X.

Algorithm 5 Classical attack to recover secret Y in the HB# protocol

Require: partial key $X \in \mathbb{Z}_2^{n_X \times m}$; algorithms $\mathcal{A}, \mathcal{B}$ (Alg. 7, 8); oracle $\mathcal{O}_P$; oracle $\mathcal{O}_V$
Ensure: $R = Y$

$\mathbf{a} \leftarrow (0, \dots 0) \in \mathbb{Z}_2^{n_X}$
$B \leftarrow \underline{\mathbf{0}} \in \mathbb{Z}_2^{n_Y \times n_Y}$ ▷ Zero matrix
$C \leftarrow \underline{\mathbf{0}} \in \mathbb{Z}_2^{n_Y \times m}$ ▷ Zero matrix
$i \leftarrow 1$
while $i \leq n_Y$ **do**
  $B_i \leftarrow \mathcal{O}_P^{(P \to V)}$
  if $\mathbf{rank}(B) \stackrel{?}{=} i$ **then**
    $\mathbf{a} \to \mathcal{O}_P^{(V \to P)}$
    $\mathbf{z}_i \leftarrow \mathcal{O}_P^{(P \to V)}$ ▷ $\mathbf{z}_i = B_i \cdot Y \oplus \mathbf{v}_i$
    $i \leftarrow i + 1$
  end if
  reset$(\mathcal{O}_P)$ ▷ Run P from the start
end while
$i \leftarrow 1$
while $i \leq n_Y$ **do**
  $\mathbf{z}'_i \leftarrow \mathcal{A}^{\mathcal{O}_V}(X, B_i, \mathbf{z}_i)$
  $\mathbf{v}'_i \leftarrow \mathcal{B}^{\mathcal{O}_V}(X, B_i, \mathbf{z}'_i)$
  $C_i \leftarrow \mathbf{z}'_i \oplus \mathbf{v}'_i$ ▷ $C_i = B_i \cdot Y$
  $i \leftarrow i + 1$
end while
$R \leftarrow B^{-1}C$
return R

This is done by interacting with the Prover via $\mathcal{O}_P^{(i)}$ to recover $X_i \in \mathbb{Z}_2^{n_X}$, for $1 \leq i \leq m$, and construct $X \in \mathbb{Z}_2^{n_X \times m}$.

Stage 2: recovery of Y. Similar to the HB+ protocol attack, stage 2 of our key-recovery attack against HB# requires the adversary to interact, classically only, first with the honest Prover P and then with Verifier V. However, because the HB# protocol runs in a single round, recovery of the secret $Y \in \mathbb{Z}_2^{n_Y \times m}$ is slightly more elaborate. The attack is described in Algorithm 5, and makes use of two subroutines that essentially run a version of a "reaction attack", similar to as described in [HGS99].

These two algorithms, while based on [HGS99], have slight adaptations: In [HGS99], the algorithms are used to, e.g., recover the message from a McEliece ciphertext $c = G \cdot m + e$, where $\mathrm{wt}(e) = t$. Their Algorithm $\mathcal{A}$ produces a new ciphertext $c' = G \cdot m + e'$, where $\mathrm{wt}(e') = t + 1$. Then, Algorithm $\mathcal{B}$ does bit-flips, bit-by-bit, to identify the positions of those $t + 1$ errors, to recover error-free $G \cdot m$, and thus m.

In our setting, the interaction occurs in multiple phases, as we attack interactive ID schemes, with strings $\mathbf{z}_i = \mathbf{b}_i \cdot Y \oplus \mathbf{v}_i$, where $\mathrm{wt}(\mathbf{v}_i)$ is at most $\tau' m$. More specifically, we use two algorithms: Algorithm $\mathcal{A}$ (described in Algorithm 7 in Appendix B), which given a noisy codeword and an oracle that is able to decode at most δ errors (in our case $\delta = \tau' m$), produces a modified noisy codeword with exactly $\delta + 1$ errors; and Algorithm $\mathcal{B}$ (Algorithm 8 in Appendix B), which given a noisy codeword with $\delta + 1$ errors and an oracle that is able to decode at most δ errors, recovers the error vector. In our case, we will make use of the Verifier V as an oracle $\mathcal{O}_V$, which will check whether the response to a challenge $\mathbf{b}$ is valid, essentially checking the number of errors of a noisy codeword (i.e. the Hamming weight of the noise vector). This will allow the adversary to remove the errors for a sufficient number of challenges, and recover the secret.

In more detail, the attack can be described as follows (Algorithms 7 and 8 are given in Appendix B).

1. The Adversary, acting as the Verifier, repeatedly interacts with P, always choosing its challenge as $\mathbf{a} = (0, \ldots, 0) \in \mathbb{Z}_2^{n_X}$, until it has collected n_Y pairs

$$(\mathbf{b}_i, \mathbf{z}_i) = (\mathbf{b}_i, \mathbf{b}_i \cdot Y \oplus \mathbf{v}_i),$$

where the n_Y vectors $\mathbf{b}_i \in \mathbb{Z}_2^{n_Y}$ randomly generated by P are linearly independent. We note that for all these pairs, we expect to have $\mathrm{wt}(\mathbf{v}_i) \leq \tau' m$ with overwhelming probability. The number of interactions required should be very close to n_Y: with probability ≈ 0.288 the first n_Y interactions should already produce such an LI set of vectors; otherwise a few extra interactions should suffice. (See the discussion under "Correctness".)

2. For each linearly independent vector $\mathbf{b}_i$ and corresponding $\mathbf{z}_i = \mathbf{b}_i \cdot Y \oplus \mathbf{v}_i$, for $i \in (1, \ldots, n_Y)$, the Adversary proceeds as follows:

   (a) Use Algorithm 7 (based on Algorithm $\mathcal{A}$ of [HGS99]) to interact with an honest Verifier V, and manipulate $\mathbf{z}_i$ to obtain the vector $\mathbf{z}'_i$,

$$\mathbf{z}'_i = \mathbf{b}_i \cdot Y \oplus \mathbf{v}'_i,$$

   where $\mathrm{wt}(\mathbf{v}'_i) = \tau' m + 1$. That means that, if $\mathbf{a}$ is a challenge from V, then $\mathbf{z}'_i \oplus \mathbf{a} \cdot X$ is *exactly* one error away from being accepted by V in a successful authentication attempt. In Algorithm 7, essentially we have V acting as an oracle for the maximum number of errors in a query. Note that the adversary knows the secret X, and can thus remove the influence of the fresh challenges from V. This procedure requires in the worst case $2(\tau' m) + 1 \leq m + 1$ queries to V.

   (b) With $\mathbf{z}'_i$, the adversary can perform Algorithm 8 (equivalent to Algorithm $\mathcal{B}$ of [HGS99]), running through the m bits, flipping each individual bit, and interacting with V. Successful authentication on flipping the j-th bit of $\mathbf{z}'_i$ means that the j-th bit of $\mathbf{v}'_i$ is 1. This requires at most $m - 1$ queries to V.

   (c) After recovering $\mathbf{v}'_i$, the adversary can compute $\mathbf{c}_i = \mathbf{b}_i \cdot Y = \mathbf{z}'_i \oplus \mathbf{v}'_i \in \mathbb{Z}_2^m$.

3. The secret $Y \in \mathbb{Z}_2^{n_Y \times m}$ can then be computed as

$$Y = B^{-1} \cdot C,$$

where C is the $n_Y \times m$ matrix with $\mathbf{c}_i$ as its rows, and B the $n_Y \times n_Y$ invertible matrix with $\mathbf{b}_i$ as rows.

Stage 2 of our HB# attack essentially runs a version of "reaction attack", as described in [HGS99]. The attack to recover Y requires approximately n_Y classical queries to P, and at most $n_Y(m + 1 + m - 1) = 2mn_Y$ classical queries to V. That is, we expect that with at most $(2m + 1)n_Y$ classical queries, the adversary is able to recover Y.

Correctness. First note that at each round i of Algorithm 5, $\mathbf{rank}(B) = i$, which is true since the probability that $\mathcal{O}_P^{P \to V}$ outputs a linearly independent B_i is large. This follows from the fact that, given k linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{Z}_2^{n_Y}$, the probability that an additional independently sampled random vector $\mathbf{b}_{k+1}$ does not lie in the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_k$ is $p_k = 1 - 2^{k - n_Y}$. Hence, using the chain rule of conditional probability, $\prod_{k=0}^{n_Y - 1} p_k \approx 0.288$ for $n_Y > 10$. Then, assuming the correctness of Algorithms 7,8 [HGS99], $C_i = B_i \cdot Y$, and therefore after n_Y rounds $C = B \cdot Y$, where B is square and $\mathbf{rank}(B) = n_Y$.

7.4 Impersonation attacks

The protocols in the HB-family provide only one-party authentication — only the Prover authenticates itself to the Verifier. Thus, a successful attack defeating the protocol only requires that the adversary is able to impersonate the Prover to an honest Verifier. Although the full key-recovery attacks described above are very efficient, we note that they require the adversary to interact (quantumly) with the Prover (to recover the secret $\mathbf{x}$ or X), and then (classically) with the Verifier (to recover $\mathbf{y}$ or Y). However, some attack models allow the attacker to interact only with the Prover, and then assume they will have one opportunity to attempt a successful authentication with the Verifier.

It is not hard to see that recovery of the secret $\mathbf{x}$ (resp. X) in the quantum part of the attacks already defeats the security of the HB+ (resp. HB#) protocol, by allowing the adversary to indefinitely impersonate the Prover to a Verifier: this can be done by always setting $\mathbf{b}$ as the all-zero vector, and when challenged by an honest Verifier with $\mathbf{a}$, simply responding with $z = \mathbf{a} \cdot \mathbf{x}$ (resp. $\mathbf{z} = \mathbf{a} \cdot X$), which will be obviously accepted by the Verifier, and result in a successful authentication.

This active attack, with partial key-recovery leading to unlimited impersonation, requires (quantum) interaction *only* with the Prover. We note that this impersonation attack can also be used to defeat the security of the Trusted-HB protocol [BC08].

8 Discussions and Future Work

The mainstream approach for the security assessment of cryptographic algorithms and schemes against quantum adversaries is currently mostly restricted to the PQC setting: one assumes the adversary has access to a large, fault-tolerant quantum computer that can be used to *quantumly* solve problems that are considered intractable for classical computers. In this setting the focus of a quantum cryptanalyst is on studying quantum algorithms that may offer substantial speed-ups compared to their classical counterparts.

However, with the advent of quantum networks, we argue for the importance of assessing cryptographic schemes also under more powerful quantum attacks. Besides the natural questions of theoretical interest, of whether the design's security guarantees may hold under extended adversarial models, the possibility of implementations of existing cryptographic algorithms and protocols as part of quantum-enabled networks — either deliberately or inadvertently — may also pose a realistic threat in the quantum era, as adversaries may be able to interact with the algorithms not only offline with a quantum computer (*Q1* model), but also online with messages in superposition (*Q2* model).

Our attacks against the HB authentication protocols under the latter model demonstrate that strong, provable security guarantees in the classical setting do not necessarily translate to quantum security. The attack technique, based on the Bernstein-Vazirani algorithm, is (in hindsight) conceptually simple, perhaps not in dissimilar fashion to the pioneering quantum superposition attacks by Kuwakado and Morii [KM10, KM12]. But they are also, to the best of our knowledge, the first superposition attacks against interactive authentication protocols — in contrast to, say, standalone symmetric-key algorithms — which in our opinion make the results of more relevance in the context of quantum networks.

As for our choice of schemes to illustrate our attack technique, namely a family of LPN-based symmetric-key authentication protocols, we note that while they were proposed for use in a very different environment — identification/authentication mechanisms for lightweight applications — we believe that their features based on simple linear operations could make these types of protocols potential candidates for consideration for deployment in emerging quantum-enabled networks. For example, we have recently seen the first demonstration of multi-node quantum networks distributing long-lived entanglement [PHB+21]. Analogous to the Noisy Intermediate Scale Quantum Computers (NISQ) [Pre18], the

network nodes in near-term quantum networks will be noisy and communicate at low rates. For these networks, low complexity quantum identification methods will be necessary. In the noisy storage model, identification methods compatible with commercial quantum key distribution hardware are known [DFSS07, Sch10]. It would be useful to find simple identification protocols that are quantum safe even in future fault-tolerant quantum networks. We believe that coherent variations of protocols such as HB+/HB# may be promising due to their low computational requirements — based only on the dot product operation.

While in this work we have focused exclusively on the HB protocols, we note that other LPN-based authentication protocols have also been proposed, for example, [HKL+12, KPV+17]. We have not considered these closely, and as further work it may be interesting to investigate whether the techniques introduced here may apply to these protocols, as well as to other LPN-based constructions, e.g. symmetric encryption algorithms.

A main contribution of our work is to show that the currently popular design principle for cryptographic schemes, where complexity is based on secret linear functions perturbed by errors, can be completely insecure in the superposition query model. Indeed, our attacks are made possible because the Bernoulli-sampled error values are translated into a global phase, allowing the application of the Bernstein-Vazinari method to recover the secret linear component. However we note that the specification of the HB protocols does not necessarily rely on having the errors being generated by a Bernoulli sampler. Acceptance by the Verifier follows from checking the Hamming weight of the error vector, and vectors satisfying a certain weight bound could well be generated based on another method.

For example, the protocols could be implemented based on a "deterministic" version of LPN. In these constructions, the "noise" is computed as $e_{s,a} = g_s(a)$, where $a \in \mathbb{Z}_2^n$ is the input, s is the secret, and g_s is a member of a family of (typically low-depth) Boolean functions of expected weight $(\tau \cdot 2^n)$. Of course, such a version of HB+/HB# may no longer claim to have its security directly linked to the hardness of the (standard) LPN problem. However, there are heuristic indications that, if the Boolean function is appropriately chosen, solving the "deterministic LPN problem" may also be a hard task (see [BCG+21]), and thus perhaps for practical purposes this variant could be an adequate alternative.

Interestingly, we note that our attacks do *not* work against the versions of the HB protocols based on deterministic LPN. In that case, the noise will depend on the input to the oracle, and therefore does not become a global phase. It would be interesting to study whether different superposition attack techniques could be applied against the deterministic LPN variants, although we would be surprised if they turned out to be as simple and efficient as the attacks we present. This is in contrast to the classical setting, where one expects that the HB protocols based on the deterministic LPN to be no more secure than the standard versions.

We present some countermeasures that may be employed to avoid the attacks from this work. A basic protection would be to not allow coherent inputs in superposition by measuring all inputs before making any computation with then, or before outputting any value. However, authentication schemes are often used as a single building block in more complex systems, and besides making systems more intricate, this would not safeguard against victims that are tricked into computing in superposition, and is yet another extra step in every protocol specification. Another possibility would be to use deterministic error values that are derived from the input together with the secret. However, this forfeits the provable security characteristic of most schemes that rely on errors being sampled from specific probability distributions. Finally, we emphasise that the main conceptual contribution of this work is to raise the concern that cryptosystems that base their security on linear functions with introduced error between possibly malicious inputs and secret values may be vulnerable and require special attention in sensitive systems that must account for powerful adversaries capable of superposition attacks.

Acknowledgements

We would like to thank the IACR CiC reviewers for their careful reading and constructive feedback. This work was partially supported by Japan's Council for Science, Technology and Innovation (CSTI) under the Cross-ministerial Strategic Innovation Promotion Program (SIP) for "Promoting the application of advanced quantum technology platforms to social issues" (grant JPJ012367). Work supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia I.P. (FCT) under projects UID/50021/2025 and UID/PRR/50021/2025, and European Defence Fund (EDF) grant agreement No. 101168112 (SEQURED).

References

[ABF+16] Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. Computational Security of Quantum Encryption. In Anderson C.A. Nascimento and Paulo Barreto, editors, *Information Theoretic Security*, pages 47–71, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-49175-2_3.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307, 2003. doi:10.1109/SFCS.2003.1238204.

[BC08] Julien Bringer and Hervé Chabanne. Trusted-HB: A Low-Cost Version of HB+ Secure Against Man-in-the-Middle Attacks. *IEEE Trans. Inf. Theory*, 54(9):4339–4342, 2008. doi:10.1109/TIT.2008.928290.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006), 29 June 2006, Lyon, France*, pages 28–33. IEEE Computer Society, 2006. doi:10.1109/SECPERU.2006.10.

[BCG+21] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Low-Complexity Weak Pseudorandom Functions in AC0[MOD2]. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 487–516, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-84259-8_17.

[BFKL94] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 278–291, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg. doi:10.1007/3-540-48329-2_24.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003. doi:10.1145/792538.792543.

[BLNS21] Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. Quantum Linearization Attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 422–452, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-92062-3_15.

[BS16] Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78(1):351–382, January 2016. doi:10.1007/s10623-015-0157-4.

[BTV16] Sonia Bogos, Florian Tramèr, and Serge Vaudenay. On solving LPN using BKW and variants. *Cryptography and Communications*, 8(3):331–369, July 2016. doi:10.1007/s12095-015-0149-2.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.

[BZ13a] Dan Boneh and Mark Zhandry. Quantum-Secure Message Authentication Codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 592–608, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38348-9_35.

[BZ13b] Dan Boneh and Mark Zhandry. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 361–379, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-40084-1_21.

[CSS15] Andrew W. Cross, Graeme Smith, and John A. Smolin. Quantum learning robust against noise. *Phys. Rev. A*, 92:012327, Jul 2015. doi:10.1103/PhysRevA.92.012327.

[CT24] André Chailloux and Jean-Pierre Tillich. The Quantum Decoding Problem. In Frédéric Magniez and Alex Bredariol Grilo, editors, *19th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2024, September 9-13, 2024, Okinawa, Japan*, volume 310 of *LIPIcs*, pages 6:1–6:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.TQC.2024.6.

[DFNS14] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition Attacks on Cryptographic Protocols. In Carles Padró, editor, *Information Theoretic Security*, pages 142–161, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-04268-8_9.

[DFSS07] Ivan B. Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Secure Identification and QKD in the Bounded-Quantum-Storage Model. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 342–359, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74143-5_19.

[DK07] Dang Nguyen Duc and Kwangjo Kim. Securing HB+ against GRS man-in-the-middle attack. In *Proc. Of SCIS 2007*, pages 23–26. Institute of Electronics, Information and Communication Engineers, 2007. URL: https://koasas.kaist.ac.kr/bitstream/10203/23125/1/SCIS2007_Duc.pdf.

[EKM17] Andre Esser, Robert Kübler, and Alexander May. LPN Decoded. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 486–514, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-63715-0_17.

[FMI+06] Marc P. C. Fossorier, Miodrag J. Mihaljević, Hideki Imai, Yang Cui, and Kanta Matsuura. An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - INDOCRYPT 2006*, pages 48–62, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11941378_5.

[FS09] Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. Cryptology ePrint Archive, Paper 2009/044, 2009. URL: https://eprint.iacr.org/2009/044.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic Security and Indistinguishability in the Quantum World. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 60–89, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg. doi:10.1007/978-3-662-53015-3_3.

[GJL14] Qian Guo, Thomas Johansson, and Carl Löndahl. Solving LPN Using Covering Codes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 1–20, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-45611-8_1.

[GKZ19] Alex B. Grilo, Iordanis Kerenidis, and Timo Zijlstra. Learning-with-errors problem is easy with quantum samples. *Phys. Rev. A*, 99:032314, Mar 2019. doi:10.1103/PhysRevA.99.032314.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[GRS05] Henri Gilbert, Matthew Robshaw, and Herve Sibert. An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol. *IEE Electronics Letters*, 41(21):1169–1170, 2005. doi:10.1049/el:20052622.

[GRS08a] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good Variants of HB$^+$ Are Hard to Find. In Gene Tsudik, editor, *Financial Cryptography and Data Security, 12th International Conference, FC 2008, Cozumel, Mexico, January 28-31, 2008, Revised Selected Papers*, volume 5143 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2008. doi:10.1007/978-3-540-85230-8_12.

[GRS08b] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. HB#: Increasing the Security and Efficiency of HB+. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 361–378, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-78967-3_21.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure Human Identification Protocols. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 52–66, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. doi:10.1007/3-540-45682-1_4.

[HGS99] Chris Hall, Ian Goldberg, and Bruce Schneier. Reaction Attacks against Several Public-Key Cryptosystems. In Vijay Varadharajan and Yi Mu, editors, *Information and Communication Security*, pages 2–12, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/978-3-540-47942-0_2.

[HKL+12] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An Efficient Authentication Protocol Based on Ring-LPN. In Anne Canteaut, editor, *Fast Software Encryption*, pages 346–365, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-34047-5_20.

[JW05] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 293–308, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11535218_18.

[KLLNP16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum Differential and Linear Cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2016(1):71–94, Dec. 2016. doi:10.13154/tosc.v2016.i1.71-94.

[KM10] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *2010 IEEE International Symposium on Information Theory*, pages 2682–2685, 2010. doi:10.1109/ISIT.2010.5513654.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *2012 International Symposium on Information Theory and its Applications*, pages 312–316, 2012. URL: https://ieeexplore.ieee.org/document/6400943.

[KPV+17] Eike Kiltz, Krzysztof Pietrzak, Daniele Venturi, David Cash, and Abhishek Jain. Efficient Authentication from Hard Learning Problems. *Journal of Cryptology*, 30(4):1238–1275, October 2017. doi:10.1007/s00145-016-9247-3.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and Concurrent Security of the HB and HB+ Protocols. *Journal of Cryptology*, 23:402–421, 2010. doi:10.1007/s00145-010-9061-2.

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. Quantum Information Set Decoding Algorithms. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography*, pages 69–89, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-59879-6_5.

[LF06] Éric Levieil and Pierre-Alain Fouque. An Improved LPN Algorithm. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, pages 348–359, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11832072_24.

[MP07] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Comput. Networks*, 51(9):2262–2267, 2007. doi:10.1016/J.COMNET.2007.01.011.

[Nat] National Institute of Standards and Technology (NIST). Post-Quantum Cryptography Standardization webpage. Accessed: 2025-05-25. URL: https://csrc.nist.gov/pqc-standardization.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. doi:10.1017/CBO9780511976667.

[OOV08] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, pages 108–124, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-89255-7_8.

[PHB+21] Matteo Pompili, Sophie LN Hermans, Simon Baier, Hans KC Beukers, Peter C Humphreys, Raymond N Schouten, Raymond FL Vermeulen, Marijn J Tiggelman, Laura dos Santos Martins, Bas Dirkse, et al. Realization of a multinode quantum network of remote solid-state qubits. *Science*, 372(6539):259–264, 2021. doi:10.1126/science.abg1919.

[Pie12] Krzysztof Pietrzak. Cryptography from Learning Parity with Noise. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science*, pages 99–114, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-27660-6_9.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. doi:10.22331/q-2018-08-06-79.

[Sch10] Christian Schaffner. Simple protocols for oblivious transfer and secure identification in the noisy-quantum-storage model. *Phys. Rev. A*, 82:032308, Sep 2010. doi:10.1103/PhysRevA.82.032308.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700.

[Sim94] D.R. Simon. On the power of quantum computation. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994. doi:10.1109/SFCS.1994.365701.

[SOKB25] Alexander Schmidhuber, Ryan O'Donnell, Robin Kothari, and Ryan Babbush. Quartic Quantum Speedups for Planted Inference. *Phys. Rev. X*, 15:021077, Jun 2025. doi:10.1103/PhysRevX.15.021077.

[XY19] Huiqin Xie and Li Yang. Using Bernstein-Vazirani algorithm to attack block ciphers. *Designs, Codes and Cryptography*, 87(5):1161–1182, May 2019. doi:10.1007/s10623-018-0510-5.

[XY20] Huiqin Xie and Li Yang. A quantum related-key attack based on the Bernstein-Vazirani algorithm. *Quantum Information Processing*, 19(8):240, July 2020. doi:10.1007/s11128-020-02741-2.

[Zha12] Mark Zhandry. How to Construct Quantum Random Functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, 2012. doi:10.1109/FOCS.2012.37.

[ZY21] Bao-Min Zhou and Zheng Yuan. Quantum key-recovery attack on Feistel constructions: Bernstein–Vazirani meet Grover algorithm. *Quantum Information Processing*, 20(10):330, September 2021. doi:10.1007/s11128-021-03256-0.

A Stage 1 of HB+ Attack

Algorithm 6 Quantum attack to recover secret $\mathbf{x}$ in the HB+ protocol

Require: oracle $\mathcal{O}_P$; quantum register $|\varphi\rangle \in \mathbb{C}^{2^n}$; classical register $\mathbf{r} \in \mathbb{Z}_2^n$
Ensure: $\mathbf{r} = \mathbf{x}$
1: $|\varphi\rangle \leftarrow |0\rangle^{\otimes n}$
2: $|\varphi\rangle \leftarrow H^{\otimes n}|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{a}\in\mathbb{Z}_2^n} |\mathbf{a}\rangle$
3: $|\varphi\rangle \leftarrow \mathcal{O}_P|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}\oplus e} |\mathbf{a}\rangle$
4: $|\varphi\rangle \leftarrow H^{\otimes n}|\varphi\rangle = |\mathbf{x}\rangle$
5: $\mathbf{r} \leftarrow$ **measure**$(|\varphi\rangle)$
6: **return r**

Correctness: In Step 3, the Prover P is run via the oracle $\mathcal{O}_P$. Moreover, in Step 4 the outcome of the computation is $|\mathbf{x}\rangle$ because

$$\begin{aligned} \left|\varphi^{(4)}\right\rangle &= \frac{1}{\sqrt{2^n}} H^{\otimes n} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}\oplus e} |\mathbf{a}\rangle \right) \\ &= (-1)^{\mathbf{b}\cdot\mathbf{y}\oplus e} \frac{1}{\sqrt{2^n}} H^{\otimes n} \left(\sum_{\mathbf{a}\in\mathbb{Z}_2^n} (-1)^{\mathbf{a}\cdot\mathbf{x}} |\mathbf{a}\rangle \right) \\ &= (-1)^{\mathbf{b}\cdot\mathbf{y}\oplus e} |\mathbf{x}\rangle \\ &= |\mathbf{x}\rangle\,. \end{aligned} \tag{16}$$

Here, $(-1)^{\mathbf{b}\cdot\mathbf{y}\oplus e}$ is a global phase, since $\mathbf{b}\cdot\mathbf{y}\oplus e$ is independent of the secret $\mathbf{x}$ and input $\mathbf{a}$. Thus, the resulting physical state is $|\mathbf{x}\rangle$, which allows the recovery of the secret $\mathbf{x} \in \mathbb{Z}_2^n$ by measuring the n qubits of $|\varphi\rangle$ in the computational basis.

B Stage 2 of HB# Attack: Algorithms from [HGS99]

We describe below two algorithms, based on Algorithm $\mathcal{A}$ and Algorithm $\mathcal{B}$ of [HGS99] respectively, that are used in the key-recovery attack against the HB# protocol.

Algorithm 7 Determine codeword with exactly $\tau' m + 1$ errors (based on Algorithm $\mathcal{A}$ [HGS99])

Require: partial key X; $\mathbf{b}$; $\mathbf{z} = \mathbf{b}\cdot Y \oplus \mathbf{v}$; oracle $\mathcal{O}_V$
Ensure: $\mathbf{z}' = \mathbf{b}\cdot Y + \mathbf{v}' :\ \mathrm{wt}(\mathbf{v}') = \tau' m + 1$
1: **for** $i = 1$ to m **do**
2: $\quad \mathbf{z}' \leftarrow \mathbf{z}$
3: $\quad$ **for** $j = 1$ to i **do**
4: $\quad\quad z'_j \leftarrow z'_j \oplus 1$
5: $\quad$ **end for**
6: $\quad \mathbf{b} \rightarrow \mathcal{O}_V^{(P\rightarrow V)}$
7: $\quad \mathbf{a} \leftarrow \mathcal{O}_V^{(V\rightarrow P)}$
8: $\quad \mathbf{z}' \oplus \mathbf{a}\cdot X \rightarrow \mathcal{O}_V^{(P\rightarrow V)}$
9: $\quad$ **if** $\mathcal{O}_V$ rejects **then**
10: $\quad\quad$ **return** $\mathbf{z}'$
11: $\quad$ **end if**
12: **end for**

Algorithm 8 Determine error vector with exactly $\tau' m + 1$ errors (based on Algorithm $\mathcal{B}$ [HGS99])

Require: partial key X; $\mathbf{b}$; $\mathbf{z}' = \mathbf{b} \cdot Y \oplus \mathbf{v}' : \mathrm{wt}(\mathbf{v}') = \tau' m + 1$; oracle $\mathcal{O}_V$
Ensure: $\mathbf{v}'' = \mathbf{v}'$

$\mathbf{v}'' \leftarrow (0, \ldots, 0) \in \mathbb{Z}_2^m$
for $j = 1$ to $m - 1$ **do**
  $\mathbf{z}'' \leftarrow \mathbf{z}'$
  $z_j'' \leftarrow z_j'' \oplus 1$
  $\mathbf{b} \rightarrow \mathcal{O}_V^{(P \rightarrow V)}$
  $\mathbf{a} \leftarrow \mathcal{O}_V^{(V \rightarrow P)}$
  $\mathbf{z}'' \oplus \mathbf{a} \cdot X \rightarrow \mathcal{O}_V^{(P \rightarrow V)}$
  if $\mathcal{O}_V$ accepts **then**
    $v_j'' \leftarrow 1$
  end if
  if $\mathrm{wt}(\mathbf{v}'') \stackrel{?}{=} \tau' m + 1$ **then**
    return $\mathbf{v}''$
  end if
end for
$v_m'' \leftarrow 1$
return $\mathbf{v}''$