Proof of Plaintext Knowledge for Code-Based Public-Key Encryption Revisited

Rong Hu*, Kirill Morozov† and Tsuyoshi Takagi‡

Abstract

In a recent paper at Asiacrypt'2012, Jain et al point out that Véron code-based identification scheme is not perfect zero-knowledge. In particular, this creates a gap in security arguments of proof of plaintext knowledge (PPK) and verifiable encryption for the McEliece public key encryption (PKE) proposed by Morozov and Takagi at ACISP'2012. We fix the latter result by showing that PPK for the code-based Niederreiter and McEliece PKE's can be constructed using Stern zero-knowledge identification scheme, which is unaffected by the above mentioned problem. Since code-based verifiable encryption uses PPK as a main ingredient, our proposal presents a fix for the McEliece verifiable encryption as well. In addition, we present the Niederreiter verifiable encryption.

1 Introduction

Postquantum cryptography encompasses schemes which remain secure even against attacks using quantum computers. Two important candidates for the postquantum public key encryption are code-based public key encryption (PKE) schemes by McEliece [25] and Niederreiter [27]. Their security is based on hardness of decoding, which is a well-studied cryptographic assumption [7, 31, 15, 8, 17, 9, 3].

It is important to develop code-based cryptographic protocols in order to introduce a wide spectrum of secure services which cryptography can offer. Quite a few of such protocol already exists such as identification or digital signatures – see [15, 12] for surveys, and also [29] for related results.

In this paper, we focus on *proof of plaintext knowledge (PPK)* for code-based PKE. Suppose that a prover P encrypted the plaintext m into ciphertext c on the public key pk. Now, PPK allows P to convince a verifier V, who does not have the secret key, that P knows m. We consider zero-knowledge (ZK) proofs which means that PPK does not leak any information on m to V.

Verifiable encryption with respect to some binary relation R on the plaintexts is a zero-knowledge proof on public inputs pk, c, and δ that allows a prover P to convince a verifier V that c is a ciphertext of m under pk such that $(m, \delta) \in R$.

1.1 Related Works

Proof of Plaintext Knowledge. PPK were first introduced by Aumann and Rabin [1] in the generic case of any PKE, and then investigated by Katz [22], who presented efficient PPK for RSA, Rabin, ElGamal and Paillier cryptosystems using Sigma-protocols. The first PPK in postquantum setting for the lattice-based Ajtai-Dwork PKE was presented by Goldwasser and Kharchenko [20], and then a number of works for lattice-based systems followed [36, 37, 5]. It is worth noting that the works by Xagawa et al [36] and Xagawa and Tanaka [37] use a modification of Stern code-based ZK identification scheme [34]. The latter scheme was also used by Kobara et al [23] for a similar purpose in code-based oblivious transfer.

Recently, Morozov and Takagi [26] presented PPK and verifiable encryption for the McEliece PKE using code-based Véron identification scheme [35], which is, in a sense, the dual of Stern scheme [34]. However, Jain et al [21] pointed out a gap in the proof of zero-knowledge property of Véron's scheme. This also created a gap in the proofs of both primitives in [26].

*Graduate School of Mathematics, Kyushu University, Japan. E-mail: r-hu@math.kyushu-u.ac.jp
†Institute of Mathematics for Industry, Kyushu University, Japan. E-mail: morozov@imi.kyushu-u.ac.jp
‡Institute of Mathematics for Industry, Kyushu University, Japan. E-mail: takagi@imi.kyushu-u.ac.jp

It is also worth noting that Jain et al [21] presented efficient commitments and ZK proofs based on a variant of the learning parity with noise (LPN) problem (for details about this problem, see e.g. [29]).

Verifiable Encryption. Verifiable encryption was first introduced by Stadler [33] as a tool for publicly verifiable secret sharing, and later generalized by Asokan et al [2] and applied to fair exchange of digital signatures. See the works by Camenisch and Damgård [10] and by Camenisch and Shoup [11] for further developments on this topic. Verifiable encryption for Ajtai-Dwork PKE was presented in [20].

1.2 Our Contribution and Discussion

In this work, we show that the basic idea presented in [26] is valid by introducing a (zero-knowledge) PPK for Niederreiter and McEliece PKE using Stern ZK identification scheme [34]. We suggest secure parameter sets and estimate the performance of our proposal. In particular, for 80-bit equivalent security, our protocol requires 9.1 kilobytes of communication, in 28 rounds, which seems to be practical.

One can imagine various applications, where the receiver of the ciphertext might want to ensure in advance that the sender knows the plaintext inside it. For instance, in case of an auction, when the encrypted bid is sent, an adversary might intercept it and re-send it on his own behalf, in attempt to bring the auction to a draw. Clearly, deployment of PPK would prevent the above attack. One might argue that an authentication mechanism such as digital signature might be a cheaper solution, however it depends on a particular application, since PPK does *not* bind the bidder to the bid. Therefore, the interactive nature of PPK would allow the bidder to authenticate the bid, but deny it later for privacy reasons.

It is worth noting that our result implies interactive code-based IND-CCA1 PKE in the standard model [18, 19, 22].

Throughout our work, we assume the presence of the public key infrastructure (or any other key authentication mechanism), which ensures validity of receiver's public key to be used in PPK.

As in [26], our protocol works also in the standard model, as any commitment scheme can be used. In particular, one may use (efficient) computationally hiding and statistically binding commitment scheme based on hardness of syndrome decoding from [14] or [21].

The type of zero-knowledge which we obtain, being statistical or computational, depends solely on the employed commitment scheme. In case of [14, 21], which are perfectly binding and computationally hiding, the ZK property holds in computational sense.

2 Preliminaries

We borrow some parts of the presentation in this section from [26].

Let us first fix some notation. Let J be an ordered subset as follows: $\{j_i, \ldots, j_m\} = J \subseteq \{1, \ldots, n\}$, then we denote a vector $(x_{j_1}, \ldots, x_{j_m}) \in \mathbb{F}_2^m$ by x_J. Similarly, we denote by M_J the submatrix of a $(k \times n)$ matrix M consisting of the columns which correspond to the indexes of J. A concatenation of matrices $X \in \mathbb{F}_2^{k \times n_0}$ and $Y \in \mathbb{F}_2^{k \times n_1}$ is written as $(X|Y) \in \mathbb{F}_2^{k \times (n_0+n_1)}$, and for $k = 1$ this will denote concatenation of vectors. We denote by $x \xleftarrow{\$} \mathcal{X}$ a uniformly random selection of an element from its domain $\mathcal{X}$. A set of $(n \times n)$ permutation matrices is denoted by $\mathcal{S}_n$.

We denote by $\langle A(a), B(b)\rangle(c)$ a random variable representing the output of a Turing machine B following an execution of an interactive two-party protocol between a Turing machine A with private input a and a Turing machine B with private input b on a joint input c, where A and B have uniformly distributed random tapes. If a party, say A, has no input, then we omit it by writing just A (instead of $A(a)$) in the above notation.

In our two-party protocols, we will denote an honest prover by P and an honest verifier by V, while a dishonest party will be denoted by $\widetilde{\mathsf{P}}$ and $\widetilde{\mathsf{V}}$, respectively.

We call a function $\epsilon(n)$ *negligible in some parameter* n, if $\epsilon(n) = 2^{-\omega(\log n)}$. We call a probability $1 - \epsilon(n)$ *overwhelming*, when $\epsilon(n)$ is negligible. Occasionally, we may omit mentioning of the security parameter. In these cases, by saying that a quantity is negligible (overwhelming), we mean that it is *negligible (overwhelming) in the security parameter*. The Hamming weight of $x \in \mathbb{F}_2^n$ is denoted as $w_H(x)$.

2.1 Security Assumptions

Definition 1. *Syndrome Decoding (SD) Problem.*
Input: $H \xleftarrow{\$} \mathbb{F}_2^{(n-k)\times n}$, $y \xleftarrow{\$} \mathbb{F}_2^{n-k}$ *and* $0 < t \in \mathbb{N}$.
Output: $s \in \mathbb{F}_2^n$ *such that* $w_H(s) \leq t$, $Hs^T = y$.

This problem was shown to be NP-complete by Berlekamp et al [7]. Its equivalent dual version can be formulated as follows.

Definition 2. *General Decoding (G-SD) Problem.*
Input: $G \xleftarrow{\$} \mathbb{F}_2^{k\times n}$, $y \xleftarrow{\$} \mathbb{F}_2^n$ *and* $0 < t \in \mathbb{N}$.
Output: $x \in \mathbb{F}_2^k$, $e \in \mathbb{F}_2^n$ *s.t.* $w_H(e) \leq t$, $xG \oplus e = y$.

The following three problems use the quantities defined in the next subsection. No polynomial-time algorithm is known for these problems [15, 17, 9].

Definition 3. *Niederreiter Problem.*
Input: A public key of Niederreiter cryptosystem (H^{pub}, t)*, where* $H^{pub} \in \mathbb{F}_2^{(n-k)\times n}$, $0 < t \in \mathbb{N}$*; and a Niederreiter ciphertext* $c \in \mathbb{F}_2^n$.
Output: $m \in \mathbb{F}_2^k$ *such that* $d_H(H, M, P, s) = e$.

Definition 4. *McEliece Problem.*
Input: A public key of McEliece cryptosystem (G^{pub}, t),
where $G^{pub} \in \mathbb{F}_2^{k\times n}$, $0 < t \in \mathbb{N}$*; and a McEliece ciphertext* $c \in \mathbb{F}_2^n$.
Output: $m \in \mathbb{F}_2^k$ *such that* $d_H(mG^{pub}, c) = t$.

Definition 5. *Goppa Code Distinguishing Problem.*
Input: $H \in \mathbb{F}_2^{(n-k)\times n}$.
Decide: If H *is a parity check matrix of an* (n, k) *irreducible Goppa code, or of a random* (n, k)*-code?*

2.2 Niederreiter Cryptosystem

For a survey on the material presented in this and in the next subsections, we refer the reader to the paper by Engelbert et al [15].

The Niederreiter PKE consists of the following triplet of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with system parameters $n, t \in \mathbb{N}$:
• Key generation algorithm $\mathcal{K}$: On input n, t, generate the following matrices:
– $H \in \mathbb{F}_2^{(n-k)\times n}$ – the parity check matrix of an irreducible binary Goppa code which can correct up to t errors. Its decoding algorithm is denoted as $\mathsf{Dec}_{\mathcal{H}}$.
– $M \in \mathbb{F}_2^{(n-k)\times(n-k)}$ – a random non-singular matrix.
– $P \in \mathbb{F}_2^{n\times n}$ – a random permutation matrix.
– $H^{pub} = MHP \in \mathbb{F}_2^{(n-k)\times n}$.

Output the public key $pk = (H^{pub}, t)$ and the secret key $sk = (M, H, P, \mathsf{Dec}_{\mathcal{H}})$.
• Encryption algorithm $\mathcal{E}$: On input a plaintext $m \in \mathbb{F}_2^n$ such that $w_H(m) = t$, and the public key pk, output the ciphertext $c = H^{pub}m^T$.
• Decryption algorithm $\mathcal{D}$: On input a ciphertext c and the secret key sk, calculate:
– $M^{-1}c = (HP)m^T$.
– Since $(HP)m^T = H(Pm^T)$, use the syndrome decoding algorithm $\mathsf{Dec}_{\mathcal{H}}$ to recover Pm^T.
–Output $m^T = P^{-1}Pm^T$.

It is easy to check correctness of the decryption algorithm: After in the first step of decryption, we obtain a syndrome of the permuted plaintext Pm^T. Since the the decoding algorithm $\mathsf{Dec}_{\mathcal{H}}$ is known, it is easy to recover the plaintext.

We note that the plaintext space of Niederreiter PKE is the set of weight-t binary vectors. For representation of arbitrary binary vectors (of an appropriate length) as valid plaintexts, see the work by Cover [13] and its improvements by Sendrier [32].

2.3 McEliece Cryptosystem

The McEliece PKE consists of the following triplet of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with system parameters $n, t \in \mathbb{N}$:
• Key generation algorithm $\mathcal{K}$: On input n, t, generate the following matrices:
– $G \in \mathbb{F}_2^{k\times n}$ – the generator matrix of an irreducible binary Goppa code correcting up to t errors. Its decoding algorithm is denoted as $\mathsf{Dec}_{\mathcal{G}}$.

– $S \in \mathbb{F}_2^{k\times k}$ – a random non-singular matrix.
– $P \in \mathbb{F}_2^{n\times n}$ – a random permutation matrix (of size n).
– $G^{pub} = SGP \in \mathbb{F}_2^{k\times n}$.

Output the public key $pk = (G^{pub}, t)$ and the secret key $sk = (S, G, P, \mathsf{Dec}_{\mathcal{G}})$.

• Encryption algorithm $\mathcal{E}$: On input a plaintext $m \in \mathbb{F}_2^k$ and the public key pk, choose a vector $e \in \mathbb{F}_2^n$ of weight t at random, and output the ciphertext $c = mG^{pub} \oplus e$.

• Decryption algorithm $\mathcal{D}$: On input c and the secret key sk, calculate:
– $cP^{-1} = mSG \oplus eP^{-1}$.
– $mSG = \mathsf{Dec}_{\mathcal{G}}(cP^{-1})$.
– Let $J \subseteq \{1, \ldots, n\}$ be s.t. G_J is invertible.

Output $m = (mSG)_J(G_J)^{-1}S^{-1}$.

It is easy to check that the decryption algorithm correctly recovers the plaintext: Since in the first step of decryption, the permuted error vector eP^{-1} is again of weight t, the decoding algorithm $\mathsf{Dec}_{\mathcal{G}}$ successfully corrects these errors in the next step.

2.4 Proof of Plaintext Knowledge

We use the definition of [22]. For a PKE scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, denote by $c = \mathcal{E}_{pk}(m; R)$ a ciphertext of a plaintext m under public key pk using randomness R. We will call (m, R) a *witness* to the decryption of c under pk. Informally, in a PPK protocol, a sender P proves to a receiver V the knowledge of a witness to the decryption for some ciphertext c under the known public key pk.

Definition 6. *Let $\Pi = (\mathsf{P}, \mathsf{V})$ be a tuple of PPT algorithms.* Π *is a* proof of plaintext knowledge *for encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ if the following conditions hold:*
(Completeness) *For all pk output by $\mathcal{K}(1^n)$ and all c with witness w to the decryption of c under pk, we have that $\langle \mathsf{P}(w), \mathsf{V}\rangle(pk, c) = 1$. (When V outputs 1 we say it* accepts.*)*
(Soundness) *For all pk output by $\mathcal{K}(1^n)$, all c produced under pk, and for any $\widetilde{\mathsf{P}}$, we have that $\Pr[\langle \widetilde{\mathsf{P}}, \mathsf{V}\rangle(pk, c) = 1]$ is negligible.*
(Zero-knowledge) *There exists a PPT Turing machine $\mathcal{SIM}$ (called a* simulator*) such that, for all pk output by $\mathcal{K}(1^n)$, all PPT $\widetilde{\mathsf{V}}$, and all w, the following distributions are indistinguishable:*

$$\{c = \mathcal{E}_{pk}(m; R) : \langle \mathsf{P}(w), \widetilde{\mathsf{V}}\rangle(pk, c)\},$$

$$\{c = \mathcal{E}_{pk}(m; R) : \langle \mathcal{SIM}, \widetilde{\mathsf{V}}\rangle(pk, c)\},$$

in case of statistical*, respectively* computational *indistinguishability, we call the property* statistical*, respectively* computational *zero-knowledge (ZK).*

3 PPK for Niederreiter PKE

We construct the proof of plaintext knowledge for Niederreiter encryption using Stern ZK identification scheme [34]. We take the Niederreiter public key, i.e. a permuted and scrambled parity-check matrix of an irreducible binary Goppa code correcting up to t errors as common data. The plaintext is used as witness and the ciphertext is used as public identification. Since the Niederreiter PKE is deterministic, we consider the string R representing randomness in Definition 6 as empty.

We observe that the security proof of Stern's scheme [34] does not use the fact the common code is random. Of importance are only the facts that it has a particular minimum distance, which is provided by construction of the Niederreiter public key, and that the decoding problem for it is hard. The latter is ensured by the hardness of Niederreiter Problem (see Section 2.1) which we assume to hold.

Witness: $m \in \mathbb{F}_2^n$, $w_H(m) = t$, where the parameters n and t are described in Section 2.2.

Common data: (H^{pub}, t) s.t. $H^{pub} \in \mathbb{F}_2^{(n-k)\times n}$ – the Niederreiter public key, and $c = H^{pub}m^T$ – the Niederreiter ciphertext.

Protocol 1 (Niederreiter PPK).

1. P computes $y \xleftarrow{\$} \mathbb{F}_2^n$ and $\pi \xleftarrow{\$} \mathcal{S}_n$ and sends three commitments:
   – $C_1 = Com(\pi, H^{pub}y^T)$,
   – $C_2 = Com(y\pi)$,
   – $C_3 = Com((y + m)\pi)$.
2. V sends $b \xleftarrow{\$} \{0, 1, 2\}$.

3. V performs the following checks and rejects,
   if any check fails:
   – If $b = 0$,
     – P sends y, π and opens C_1 and C_2.
     – V directly checks validity of the opened values.
   – If $b = 1$,
     – P sends $y + m, \pi$ and opens C_1 and C_3.
     – V checks validity by computing the following:
   $H^{pub}y^T = H^{pub}(y + m)^T + c$, and then verifying that the opening of C_1 is $(\pi, H^{pub}y^T + c)$, and that the opening of C_3 is $(y + m)\pi$.
   – If $b = 2$,
     – P sends $y\pi, m\pi$ and opens C_2 and C_3.
     – V checks validity of opened values by verifying that C_2 opens to $y\pi$, C_3 opens to $y\pi + m\pi$, and that $w_H(m\pi) = t$.

Denote a protocol consisting of r independent iterations of Protocol 1 by $\mathsf{PPK}(H^{pub}, c; m)$, with some appropriately chosen r.

Theorem 1. *Protocol* $\mathsf{PPK}(H^{pub}, c; m)$ *is a proof of plaintext knowledge for the Niederreiter public key encryption according to Definition 6 assuming hardness of Niederreiter Problem, in the standard model.*

Proof. We generally follow the proof of [34], but for the proof of soundness we use the argument from [35], since it is shorter[1].

Completeness. It is easy to check that P who knows the plaintext can answer all of three challenges correctly. This implies that $\langle \mathsf{P}(w), \mathsf{V} \rangle(pk, c) = 1$.

Soundness.

Lemma 1. *If* V *accepts* $\widetilde{\mathsf{P}}$*'s proof with probability at least* $(\frac{2}{3})^r + \epsilon$*, then there exists a PPT algorithm* WE *which, with overwhelming probability, computes a witness* m*.*

The proof will appear in the full version of this paper. We only note that the machine WE, constructed in the proof, finds a valid witness, hereby contradicting the hardness of the Niederreiter problem, unless the binding property of the commitment is violated. Therefore, for a cheating prover $\widetilde{\mathsf{P}}$, we must have $\Pr[\langle \widetilde{\mathsf{P}}, V \rangle(pk, c) = 1] \leq (2/3)^r + \epsilon$, which is negligible in n and r.

We emphasize that the proof does not require any additional assumptions on the Niederreiter public key (such as for instance, its indistinguishability from a random matrix), except for those made in the statement of the Niederreiter problem.

Zero-knowledge. This property guarantees that the execution of the protocol does not leak any information (in computational, or in statistical sense) to the cheating polynomial-time verifier $\widetilde{\mathsf{V}}$, who might decide on an arbitrary strategy for choosing her challenges.

Lemma 2. *Protocol 1 is computational (respectively statistical) zero-knowledge according to Definition 6, if the used commitment scheme is computationally (respectively unconditionally) hiding.*

The proof will appear in the full version of this paper.
The above two lemmas now conclude the proof of Theorem 1. □

4 Performance Evaluation

In this section, we estimate security and performance of the proposed scheme. We chose the parameters of the Niederreiter scheme according to the estimation by Peters [30] for the running time of the information set decoding algorithm. We first choose the code length n (as a power of 2, for convenience) and $k = n - t \log_2 n$ (again for convenience), then we find the smallest value of t which provides us with 80 or 128 bits of equivalent security. We fix the soundness failure probability in our PPK to be at most 2^{-16}, since this value is a minimal requirement in the ISO/IEC-9798-5 standard for the zero-knowledge techniques for entity authentication. Remembering that the soundness failure probability is 2/3 in each round, we will need 28 rounds in total.

We consider the original Niederreiter public key encryption, with binary irreducible Goppa codes used for generation of public keys, without any optimizations. Then, the size of the public key is $(n - k)n$ bits, the ciphertext size is n bits.

[1] Remember that the gap in the proof of [35] pointed out in [21] concerned only the proof of zero-knowledge property.

Equivalent security (bits)	80	128
Code length n	2048	4096
Code dimension k	1806	3676
Weight of error vector t	22	35
Public key size (Kbytes)	62.3	132.3
Communication (Kbytes)	9.1	16.1
Prover's computation (operations over $\mathbb{F}_2$)	$2^{24.73}$	$2^{26.52}$

Table 1: Parameters and Performance of the Proposed PPK Protocol.

In order to keep estimation simple, we assume the random oracle model, and construct commitments using idealized hash functions $h : \{0,1\}^* \to \{0,1\}^{l_c}$, taking $l_c = 160$. In order to commit to a value x (we will think of a binary representation of x), P will simply compute $h(x)$. Then, V's checks in our protocol will be performed as in [34] by computing the corresponding hash values. For instance, we will compute $C_1 = h(\pi, H^{pub}y^T)$, and when $b = 0$, V will use the values π and $H^{pub}y^T$ received from P to compute $h(\pi, H^{pub}y^T)$ and check that it is equal to C_1 received in Step 1.

We assume that the representations of y and π, respectively, are generated using pseudorandom generators with seed length $l_s = 128$. Sending seeds instead of actual values will allows us to save on communication.

The expected communication cost of our protocol is:

$$r\big(3l_c + 2 + (3l_s + 3n)/3\big).$$

we call it "expected" because in the thirm term, we average over the sizes of responses to the challenge b.

Prover's expected amount of computation, which is also an upper bound on that amount for the verifier, is as follows:

$$r\big((2n-1)(n-k)+n\big) \text{ binary operations.}$$

This is just a rough estimate of the total computation cost, since the costs of computing permutations, commitments, and pseudorandom generation – which are implementation-specific – are not included.

Our proposed recommended parameters and the resulting costs are summarized in Table 1.

We can see that the communication cost of the proposed protocol appears to be within the practical feasibility range. For instance, for the equivalent 80 bit security, it is 9.1 kilobytes. For comparison, it is 6.9 times smaller than the (non-optimized) public key size of the Niederreiter PKE, which is equal to 62.3 kilobytes for the relevant parameters.

5 PPK for McEliece PKE

One natural way to construct the proof of plaintext knowledge for the McEliece PKE [25] would be to replace the flawed Véron ZK identification scheme [35] with the ZK proofs of Jain et al [21] in the construction of [26].

A potential problem is that the scheme of [21] is based of a variant of LPN problem. This, in turn, will require us to make an additional Goppa Code Distinguishing assumption (see Sec. 2.1) on pseudorandomness of the McEliece public key. Although this assumption has not yet been disproved directly, a distinguisher for *high rate* Goppa codes was presented by Faugère et al in [16]. Therefore, we prefer to avoid this assumption whenever possible – in particular, when constructing our PPK.[2]

Our approach is to use the equivalence between McEliece and Niederreiter PKE's observed by Li et al [24]. In order to construct PPK for the McEliece ciphertext, one will first compute an "equivalent" plaintext, along with an "equivalent" public key of the Niederreiter PKE, and then use Protocol 1. In fact the equivalent public key may be pre-computed and distributed the same way as the original McEliece public key, since the equivalence is easy to verify.

[2]At the same time, we admit that deployment of the ZK proofs of [21] seems to be a prospective way for enhancing the code-based verifiable encryption of [26].

In details, suppose that the McEliece ciphertext is $c = mG^{pub} + e$, where m, G^{pub} and e are computed as described in Section 2.3. Let the equivalent Niederreiter public key H^{pub} be chosen a basis of the kernel of G^{pub}, i.e.

$$H^{pub}(G^{pub})^T = \mathbf{0}, \tag{1}$$

where $\mathbf{0}$ is the zero-vector of the appropriate length. In other words, H^{pub} is a parity check matrix of the linear code generated by G^{pub}. The equivalent Niederreiter ciphertext c_N is computed as a syndrome of the ciphertext c, using H^{pub} as a parity check matrix: $c_N = H^{pub}c^T = H^{pub}(mG^{pub})^T + H^{pub}e^T = H^{pub}e^T$.

Let us now briefly argue that our proposal indeed works. First of all, it is easy to check that H^{pub} is chosen a basis of the kernel of G^{pub} since Equation (1) and the ranks of these matrices are easy to verify. Then, note that multiplication of H^{pub} by $(c')^T$, where c' is an arbitrary n-bit vector, will phase out any codewords of the code generated by G^{pub} from c'. Now the obtained expression is of the form $H^{pub}(e')^T$, $e' = c' + mG^{pub}$ for some $m \in \mathbb{F}_2^k$. If $w_H(e') = t$, then c' is a valid McEliece ciphertext, but also e' is a valid witness for Protocol 1. Otherwise, c' is not a valid McEliece ciphertext, but also e' is not a valid witness, and a prover must fail the PPK by Theorem 1.

Note that so far we only explained that Protocol 1 works as a ZK proof of validity for the McEliece PKE. However, the prover, who knows an error vector of appropriate weight, can compute the plaintext simply by solving an overdefined system of linear equations as $c + e = mG^{pub}$, where G^{pub} is public. Therefore, this is also a proof of plaintext knowledge.

In Definition 6, we would consider the string m representing the plaintext as empty.

Trivially, the same PPK will work for the IND-CPA variant of the McEliece PKE [28], where the plaintext is padded with uniform randomness.

6 Verifiable Encryption

Informally, verifiable encryption with respect to some binary relation R on the plaintexts is a ZK proof on public inputs pk, c, and δ that allows P to convince V that c is a ciphertext of m under pk such that $(m, \delta) \in R$. The simplest example is the equality relation $R_{eq} = \{(m, m') | m = m'\}$, i.e. that a given ciphertext c is an encryption of a given plaintext m under public key pk. We refer the reader to [26] and references therein for a formal definition and related results.

The work [26] presents the McEliece verifiable encryption with respect to the equality relation. In other words, P convinces V that a given message m is contained in the Randomized McEliece encryption, see Section 2.3 for description.

Informally, their protocol works as follows: First, P runs PPK on the input ciphertext $c = (r|m)G^{pub} + e = rG_0^{pub} + mG_1^{pub} + e$, where $(G^{pub})^T = \left((G_0^{pub})^T | (G_1^{pub})^T\right)$, $G_0^{pub} \in \mathbb{F}_2^{k_0 \times n}$ and $G_1^{pub} \in \mathbb{F}_2^{k_1 \times n}$ are the sub-matrices of G^{pub} corresponding to randomness and plaintext, respectively. Secondly, both players compute $c' = c + mG_1^{pub} = rG_0^{pub} + e$, hereby canceling out the plaintext, and then run PPK with c' as ciphertext and G_0^{pub} as public key.

Now, we can see that Protocol 1 can be used as PPK in the above construction. Hereby, we fix the McEliece verifiable encryption scheme of [26].

It was noted in [26], that although the resulting PPK is zero-knowledge, the fact that V learns m (together with the fact that c is a valid McEliece ciphertext) implies that $\widetilde{\mathsf{V}}$ can now produce a valid encryption $c_a = rG_0^{pub} + e + m_aG_1^{pub}$ for an arbitrary $m_a \in \mathbb{F}_2^k$. Note that although the Randomized McEliece PKE is clearly malleable, the above attack is not feasible for $\widetilde{\mathsf{V}}$ prior to the protocol execution. This is not a problem of the protocol, but rather the property of the Randomized McEliece encryption. Therefore, some authentication technique must be applied in order to avoid this attack.

The construction of Niederreiter verifiable encryption using PPK is similar. We deter its presentation to the full version of this paper.

7 Acknowledgments

The authors would like to thank the anonymous reviewers of AsiaCCS 2013 for pointing out some errata. Rong Hu was supported by China Scholarship Council. Kirill Morozov was supported by a *kakenhi* Grant-in-Aid for Young Scientists (B) 24700013 from Japan Society for the Promotion of Science.

References

[1] Y. Aumann and M.O. Rabin. A Proof of Plaintext Knowledge Protocol and Applications. Manuscript. June, 2001. Available as slides from 1998 IACR Distinguished Lecture by M.O. Rabin: `http://www.iacr.org/publications/dl/rabin98/rabin98slides.ps`.

[2] N. Asokan, V. Shoup, M. Waidner. Optimistic Fair Exchange of Digital Signatures (Extended Abstract). EUROCRYPT 1998: 591-606.

[3] A. Becker, A. Joux, A. May, A. Meurer: Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. EUROCRYPT 2012: 520-536.

[4] Mihir Bellare, Phillip Rogaway: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. ACM Conference on Computer and Communications Security 1993: 62-73.

[5] R. Bendlin, I. Damgård. Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems. TCC 2010: 201-218.

[6] D.J. Bernstein, T. Lange, C. Peters. Wild McEliece. Selected Areas in Cryptography 2010: 143-158.

[7] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems, IEEE Trans. on Inf. Theory 24, 1978: 384-386.

[8] D.J. Bernstein: Grover vs. McEliece. PQCrypto 2010: 73-80.

[9] D.J. Bernstein, T. Lange and C. Peters. Smaller Decoding Exponents: Ball-Collision Decoding. CRYPTO 2011: 743-760.

[10] J. Camenisch, I. Damgård. Verifiable Encryption, Group Encryption, and Their Applications to Separable Group Signatures and Signature Sharing Schemes. ASIACRYPT 2000: 331-345.

[11] J. Camenisch, V. Shoup: Practical Verifiable Encryption and Decryption of Discrete Logarithms. CRYPTO 2003: 126-144.

[12] Pierre-Louis Cayrel, Mohammed Meziani: Post-quantum Cryptography: Code-Based Signatures. AST/UCMA/ISA/ACN 2010: 82-99.

[13] T. Cover. Enumerative source encoding. IEEE Transactions on Information Theory, 19(1):73-77, January 1973.

[14] R. Dowsley, J. van de Graaf, J. Müller-Quade, A.C.A. Nascimento: Oblivious Transfer Based on the McEliece Assumptions. ICITS 2008: 107-117.

[15] D. Engelbert, R. Overbeck and A. Schmidt: A Summary of McEliece-Type Cryptosystems and their Security, Journal of Mathematical Cryptology, vol. 1, Walter de Gruyter, 2007: 151-199.

[16] Jean-Charles Faugère, Ayoub Gauthier-Umaña, Valérie Otmani, Ludovic Perret, and Jean-Pierre Tillich. A Distinguisher for High Rate McEliece Cryptosystems. Information Theory Workshop (ITW), 2011: 282-286.

[17] M. Finiasz, N. Sendrier: Security Bounds for the Design of Code-Based Cryptosystems. ASIACRYPT 2009: 88-105.

[18] Z. Galil, S. Haber, M. Yung: Symmetric Public-Key Encryption. CRYPTO 1985: 128-137.

[19] O. Goldreich: Foundations of Cryptography I: Basic Tools. Cambridge University Press, 2001.

[20] S. Goldwasser, D. Kharchenko: Proof of Plaintext Knowledge for the Ajtai-Dwork Cryptosystem. TCC 2005: 529-555.

[21] A. Jain, S. Krenn, K. Pietrzak, A. Tentes: Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. ASIACRYPT 2012: 663-680.

[22] J. Katz: Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications. EUROCRYPT 2003: 211-228.

[23] K. Kobara, K. Morozov, R. Overbeck: Coding-Based Oblivious Transfer. MMICS 2008: 142-156.

[24] Y. X. Li, R. H. Deng, and X. M. Wang: The Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems. IEEE Trans. Inform. Theory 40, 1994: 271-273.

[25] R.J. McEliece: A Public-Key Cryptosystem Based on Algebraic Coding Theory. Deep Space Network Progress Rep., 1978.

[26] K. Morozov, T. Takagi: Zero-Knowledge Protocols for the McEliece Encryption. ACISP 2012: 180-193.

[27] H. Niederreiter: Knapsack-type Cryptosystems and Algebraic Coding Theory. Prob. of Control and Inf. Theory, 15(2), 1986: 159-166.

[28] R. Nojima, H. Imai, K. Kobara, K. Morozov. Semantic security for the McEliece cryptosystem without random oracles. Des. Codes Cryptography 49(1-3), 2008: 289-305.

[29] K. Pietrzak: Cryptography from Learning Parity with Noise. SOFSEM 2012: 99-114.

[30] C. Peters: Information-Set Decoding for Linear Codes over F_q. PQCrypto 2010: 81-94.

[31] N. Sendrier. On the security of the McEliece public-key cryptosystem. Information, Coding and Mathematics – Proceedings of Workshop honoring Prof. Bob McEliece on his 60th birthday, Kluwer, 2002: 141-163.

[32] N. Sendrier: Encoding information into constant weight codewords, ISIT 2005: 435-438.

[33] M. Stadler: Publicly Verifiable Secret Sharing. EUROCRYPT 1996: 190-199.

[34] J. Stern: A new paradigm for public key identification. IEEE Trans. Inform. Theory 42(6), 1996: 1757-1768.
A conference version: J. Stern: A New Identification Scheme Based on Syndrome Decoding. CRYPTO 1993: 13-21.

[35] P. Véron. Improved identification schemes based on error-correcting codes. Appl. Algebra Eng. Commun. Comput. 8(1), 1996: 57-69.

[36] K. Xagawa, A. Kawachi, K. Tanaka. Proof of Plaintext Knowledge for the Regev Cryptosystems. Tech.rep. C-236, Tokyo Inst. of Technology, 2007.

[37] K. Xagawa, K. Tanaka. Zero-Knowledge Protocols for NTRU: Application to Identification and Proof of Plaintext Knowledge. ProvSec 2009: 198-213.