Coding-Based Oblivious Transfer

Kazukuni Kobara[1], Kirill Morozov[1], and Raphael Overbeck[2⋆]

[1] RCIS, AIST
Akihabara Daibiru, room 1102
1-18-13 Sotokanda, Chiyoda-ku
Tokyo 101-0021 Japan
{k-kobara,kirill.morozov}@aist.go.jp

[2] EPFL - I&C - ISC - LASEC
Station 14 - Building INF
CH-1015 Lausanne
Switzerland
overbeck@cdc.informatik.tu-darmstadt.de

Abstract. We present protocols for two flavors of oblivious transfer (OT): the Rabin and 1-out-of-2 OT based on the assumptions related to security of the McEliece cryptosystem and two zero-knowledge identification (ZKID) schemes, Stern's from Crypto '93 and Shamir's from Crypto '89, which are based on syndrome decoding and permuted kernels, respectively. This is a step towards diversifying computational assumptions on which OT – cryptographic primitive of central importance – can be based.

As a by-product, we expose new interesting applications for both ZKID schemes: Stern's can be used for proving correctness of McEliece encryption, while Shamir's – for proving that some matrix represents a permuted subcode of a given code.

Unfortunately, it turned out to be difficult to reduce the sender's security of both schemes to a hard problem, although the intuition suggests a successful attack may allow to solve some long-standing problems in coding theory.

Keywords: Oblivious transfer, coding-based cryptography, McEliece cryptosystem, permuted kernel problem.

1 Introduction

Oblivious transfer (OT) [23, 10, 28] is an important cryptographic primitive which implies secure two-party computation [12, 16]. OT guarantees a transmission from a sender to a receiver with partial erasure of the input,

⋆ Part of this work was done at the Cryptography and Computer Algebra Group, TU-Darmstadt, Germany. Part of this work was funded by the DFG.

which can happen in two manners. When the whole input is erased with some fixed probability (independently of the player's control), we have an analog of the erasure channel, or Rabin OT. When the sender has two inputs and one of them is received (while the other is erased) according to the receiver's choice (and the sender does not learn this choice), we have 1-out-of-2 OT. In fact, these two flavors of OT were shown to be equivalent [7].

A number of complexity assumptions were used to construct OT: generic, e.g., enhanced trapdoor permutations [10, 11, 14], and specific, e.g., factoring [23], Diffie-Hellman [3, 20, 1], N'th or Quadratic Residuosity and Extended Riemann Hypothesis [15].

Our contribution. We present two coding-based computationally secure constructions. The Rabin OT protocol is based on the McEliece encryption [19] where a public-key is constructed from the permuted concatenation of the standard McEliece public key and a random matrix. The receiver will construct this public key and prove its correctness using the Shamir's zero-knowledge identification (ZKID) scheme [25] based on permuted kernel problem (PKP). The sender will prove that the input is encrypted using the error vector of the appropriate weight using the Stern's ZKID scheme [26] based on general syndrome decoding.

We emphasize that these are new applications of the two ZKID schemes which can be of indepedent interest in coding-based protocols. In particular, combining McEliece encryption with Stern's ZKID yields the verifiable McEliece encryption (for verifiable encryption and its applications see, e.g., [4]).

Unfortunately, in the above protocol, even the honest-but-curious receiver can reduce the probability of erasure. We show that this can be fixed by applying the reduction [7]. In fact, we show that this reduction can be used for such a weaker version of Rabin OT. However, the whole construction becomes involved and we end up implementing 1-out-of-2 OT on the way. Hence, we present a generalization of the above protocol which implements 1-out-of-2 OT using the presented techniques.

The security of both protocols is based on the assumption related to security of the McEliece PKC – indintiguishability of permuted generating matrix of a Goppa code from random and bounded distance decoding – and, in addition, the assumptions underlying the used ZKID schemes. We also employ commitment schemes.

Both constructions share the same shortcoming: it turned out to be difficult to reduce the sender's secuirty to a hard decoding problem. Shortly speaking, the intuition suggests that a successfull attack would

require either efficient list decoding algorithm for Goppa codes, or extending those codes with random columns while still retaining a good error correcting capability.

Related Work. The work [9] presents 1-out-of-2 Bit OT protocol based exclusively on the McEliece PKC related assumptions. Its efficiency is comparable to our 1-out-of-2 String OT protocol, but it provides a stronger security guarantee for the receiver: it is unconditional as compared to computational in our case.

Organization. In Section 2, we briefly introduce our security definitions, assumptions, and the main ingredients for our constructions. The reduction from a weak version of Rabin OT to the original Rabin OT is presented in Section 3. Section 4 introduces our Rabin OT construction, while our 1-out-of-2 OT protocol is sketched in Section 5.

2 Preliminaries

In our definitions, the players are bounded to run in probabilistic polynomial time in a security parameter n.

For vectors, summation is component-wise in the corresponding field, unless stated otherwise. Computation indistinguishability is denoted by "$\stackrel{c}{=}$".

2.1 Security Definitions

Informally, Rabin (string) oblivious transfer is the trusted erasure channel from the sender Sen to the receiver Rec with fixed erasure probability $\mathcal{Q}^H = 1 - \mathcal{P}$ and a bit-vector $\mathbf{b} \in \mathbb{F}_2^k$ as input. The malicious sender $\widetilde{\mathsf{Sen}}$ has no knowledge on the output, while the malicious receiver $\widetilde{\mathsf{Rec}}$ cannot learn the erased input.

We denote by a $View$ of the player all the messages that he sent and received during the protocol as well as his local randomness. Let the binary random variable E (which indicates the fact of erasure and whose outcome is available to Rec) is equal to 0 with probability $\mathcal{P}$. For the sake of simplicity, in the expressions for views, we omit most of the variables which are the same on both sides of equality.

Definition 2.1. *A two-party protocol is said to* securely implement Rabin OT*, if* Sen *gets as input a* k*-bit vector* $\mathbf{m}$ *and the following conditions are satisfied:*

– *Completeness: When* Sen *and* Rec *follow the protocol, if* $E = 0$*, then* Rec *outputs* $\mathbf{m}$*, otherwise he outputs "erasure".*
– *Sender's security:* $\forall \mathbf{m}' \neq \mathbf{m}, \mathbf{m}' \in \mathbb{F}_2^k$:
$View_{\widetilde{\mathsf{Rec}}}(\mathbf{m}|E = 1) \stackrel{c}{=} View_{\widetilde{\mathsf{Rec}}}(\mathbf{m}'|E = 1)$.
– *Receiver's security:* $View_{\widetilde{\mathsf{Sen}}}(E = 0) \stackrel{c}{=} View_{\widetilde{\mathsf{Sen}}}(E = 1)$.

The definition of γ-gap Rabin OT is analogous to the above, but $\widetilde{\mathsf{Rec}}$ can decrease the erasure probability from his point of view by γ. This probability is denoted by $\mathcal{Q} = 1 - \mathcal{P} - \gamma$. Let the binary random variable $\widetilde{E}$ which indicates the fact of erasure ($\widetilde{E} = 1$) or not for $\widetilde{\mathsf{Rec}}$ be equal to 1 with probability $\mathcal{Q}$.

Definition 2.2. *A two-party protocol is said to* securely implement γ-gap Rabin OT*, if Definition 2.1 holds except that the sender's security condition is replaced with:*

$$\forall \mathbf{m}' \neq \mathbf{m}, \mathbf{m}' \in \mathbb{F}_2^k : View_{\widetilde{\mathsf{Rec}}}(\mathbf{m}|\widetilde{E} = 1) \stackrel{c}{=} View_{\widetilde{\mathsf{Rec}}}(\mathbf{m}'|\widetilde{E} = 1).$$

In the other flavor of oblivious transfer, 1-out-of-2 String OT, Sen inputs two a-bit vectors $\mathbf{b}_0, \mathbf{b}_1$. Rec obtains one of them according to his choice $c \in \{0, 1\}$. $\widetilde{\mathsf{Sen}}$ is unable to learn c, while $\widetilde{\mathsf{Rec}}$ remains ignorant about $\mathbf{b}_{1-c}$.

2.2 Assumptions

The security of all schemes we present in this paper is based on the assumption, that the following problems are hard to solve in the average case:

Definition 2.3. *In the following let all matrices and vectors be over* $\mathbb{F}_q$*.*

(i) Given a $k \times n$ *matrix, decide if its row-space is within a Goppa code or was generated at random. (Goppa-code-distinguishing Problem)*
(ii) Given a (random) $[n, k]$ *code generated by the matrix* G^{pub}*, a word* $\mathbf{c}$ *and an integer* w*, find* $\mathbf{e}$ *of Hamming weight at most* w *such that* $\mathbf{c} = \mathbf{m}\mathsf{G}^{\text{pub}} + \mathbf{e}$ *for some* $\mathbf{m}$*. (General Syndrome Decoding)*
(iii) Given a (random) $[n, k, d]$ *code generated by the matrix* G^{pub}*, find a codeword of weight* $\leq w$ *in that code (Finding low weight words).*
(iv) Given a random $[n, k]$ *code and a random permuted subcode of dimension* $l < k$*, find the permutation. (Permuted Kernel Problem)*

Only Problems *(ii)* – *(iv)* are known to be $\mathcal{NP}$-hard in the general case [25, 26]. The coding theoretic problems *(ii)* and *(iii)* seem to be the hardest, if w is close to the Gilbert-Varshamov (GV) bound (see, e.g. [18, Ch. 17, Thm. 30]).

2.3 Tools

We shortly recall the main ingredients of our scheme: the McEliece PKC, the zero-knowledge identification protocols (ZKIP) by Stern and Shamir connected to coding theory, and Crepeaus's protocol for 1-out-of-2 OT based on Rabin OT.

McEliece's public key encryption scheme [19] works as follows: Upon input of the system parameters m, t, the key generation algorithm outputs the secret key consisting of three matrices: $(\mathsf{S}, \mathsf{G}, \mathsf{P})$, where $\mathsf{G} \in \mathbb{F}_2^{k \times n}$ is a canonical generator matrix of an $[n, k \geq n - mt, 2t + 1]$ binary irreducible Goppa code, $\mathsf{S} \in \mathbb{F}_2^{k \times k}$ is non-singular and $\mathsf{P} \in \mathbb{F}_2^{n \times n}$ is a permutation matrix. The corresponding public key is $(\mathsf{G}^{\text{pub}} = \mathsf{SGP}, t)$. To encrypt a message $\mathbf{m} \in \mathbb{F}_2^k$ the sender chooses a random binary vector $\mathbf{e}$ of length n and Hamming weight t and computes the ciphertext $\mathbf{c} = \mathbf{m}\mathsf{G}^{\text{pub}} + \mathbf{e}$. The secret key holder now can recover $\mathbf{m}$ from $\mathbf{c}$ using his secret key.

For properly chosen parameters, the McEliece PKC is secure [5] and there exist conversions to obtain CCA2 security [17]. For such variants, or if only random strings are encrypted, G^{pub} can be chosen to be *systematic* (i.e. with the k-dimensional identity matrix Id_k in the first k columns), as we will do in the following. This reduces space needed to store G^{pub}.

The size of the ciphertexts can be reduced to $n - k$ if the message is represented by $\mathbf{e}$. This is known as the Niederreiter PKC, compare [24]. In the latter case (e.g. if a hash of $\mathbf{e}$ serves as a random seed or key for a symmetric encryption scheme) it is sufficient to send the syndrome $\mathbf{e}(\mathsf{G}^{\text{pub}})^{\perp}$ as ciphertext, where $(\mathsf{G}^{\text{pub}})^{\perp}$ refers to the systematic check matrix of G^{pub}.

Stern's ZKIP [26] has a check matrix $\mathsf{H} \in \mathbb{F}_q^{n \times (n-k)}$ and an integer w as system parameters. An user's identity $\mathbf{s}$ is computed from the user's secret, a vector $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight w: $\mathbf{s} = \mathbf{e}\mathsf{H}$. By Stern's 3-round zero-knowledge protocol, the secret key holder can prove his knowledge of $\mathbf{e}$ using two blending factors: a permutation and a random vector. However, a dishonest prover not knowing $\mathbf{e}$ can cheat the verifier in the protocol with probability 2/3. Thus, the protocol has to be run several times to detect cheating provers. Computing $\mathbf{e}$ from $\mathbf{s}$ is solving Problem *(ii)* from Definition 2.3. The communication cost is about $n(1 + \log_2(n)) \log_2(q)$ plus three times the size of the employed commitments (e.g. a hash function).

Shamir's Permuted Kernel ZKIP [25] works quite similarly, i.e. it has a check matrix $\mathsf{H} \in \mathbb{F}_q^{n \times (n-k)}$ and an integer l as system parameters. (Shamir proposed to use $l = 1$ and q to be a large prime. However,

taking q small and $l < (n-k)$ works as well [27].) The user's identity $\mathsf{K} \in \mathbb{F}_q^{l \times n}$ is computed from the user's secret, a permutation $\mathsf{\Pi}$ as follows: K is taken at random from the right kernel of $\mathsf{\Pi H}$. In the following we can view K as an n-vector over $\mathbb{F}_{q^l}$. By Shamir's 5-round zero-knowledge protocol, the secret key holder can prove his knowledge of $\mathsf{\Pi}$ using two blending factors: a permutation and a random n-vector over $\mathbb{F}_{q^l}$. However, a dishonest prover not knowing $\mathsf{\Pi}$ can cheat the verifier in the protocol with probability $(q^l+1)/(2 \cdot q^l)$. Thus, the protocol has to be repeated several times to detect cheating provers. Computing $\mathsf{\Pi}$ from K is solving Problem *(iv)* from Definition 2.3. The communication cost is about $n(l + \log_2(n)) \log_2(q)$ plus two times the size of the commitments. See [22] for the practical security analysis.

Crépeau's protocol [7] allows us to build an 1-out-of-2 OT from a Rabin OT and a hash function h: In a first stage, N random messages r_i are sent to the receiver by a Rabin OT with erasure (receiving) probability $\mathcal{Q}$ ($\mathcal{P}$). Now, K is chosen such that $K < \mathcal{P}N = (1-\mathcal{Q})N < 2K < N$, i.e. the receiver obtains at least K and at most $2K-1$ of the random messages r_i. Then, the receiver sends two disjoint sets $\mathcal{I}, \mathcal{J} \subseteq \{1, \ldots, N\}$ of K indices to the sender, such that one of the sets contains only indices of not erased messages r_i. For the 1-out-of-2 OT, the messages $\mathbf{m}_0, \mathbf{m}_1$ are encrypted as $\mathbf{c}_0 = \mathbf{m}_0 + h((r_i)_{i \in \mathcal{I}})$ and $\mathbf{c}_1 = \mathbf{m}_1 + h((r_j)_{j \in \mathcal{J}})$. Since the receiver knows either the set $(r_i)_{i \in \mathcal{I}}$ or $(r_j)_{j \in \mathcal{J}}$, he obtains exactly one of the messages from $\mathbf{c}_0$ and $\mathbf{c}_1$. Crépeau's protocol fails with some probability which is negligible in N and can easily be computed.

Commitment scheme. This protocol allows a committer to transmit an evidence (called *commitment*) of a message to the verifier with possibility to reveal the message later. The committer cannot learn the message before revealing, while the verifier cannot change his mind by opening a different message. In this work, we use a commitment functionality abstracting from its implementation. For details on commitment schemes, see [8] and the references therein.

3 Reducing the Gap in Rabin OT

We show that the Crépeau's protocol can be used to reduce the Gap Rabin OT to the 1-out-of-2 OT and thus to the original Rabin OT.

Theorem 3.1. *γ-Gap Rabin OT with $\gamma = \mathcal{Q}^{\mathrm{H}} - Q^A$ is equivalent to Rabin OT, if for some $K' > 0$: $K' < 1 - Q^H \leq 1 - Q^A < 2K' < 1$.*

Proof (Sketch). Consider the Crépeau's protocol as described in the previous section and take $K = K'N$. It is easy to check that this protocol

works, i.e., the sender is very likely to find enough received messages for one set, while the other set is very likely to contain at least one erasure.

4 Rabin Oblivious Transfer

Our scheme implementing Rabin OT with erasure probability $1-\mathcal{P}$ consists of two phases: initialization and transmission. The first one is used for key generation, where a Goppa code is concatenated with a random code and used as substitute for the secret code in the McEliece PKC, see Algorithm 4.1. To ensure correct generation of the public key, we use a trusted third party (TTP) in Algorithm 4.1, which can be omitted as we show in Section 4.2. In the transmission phase, see Algorithm 4.2, en- and

Algorithm 4.1 Key generation

Input: Security parameters $m, t, t', l \in \mathbb{N}$.

Receiver: Set $n = 2^m$, $k = 2^m - mt$. Generate a McEliece PKC key pair with security parameters m, t. Let $(\mathsf{S}, \mathsf{G}, \mathsf{P})$ be the secret key with public key $(\mathsf{G}^{\text{pub}} = \mathsf{SGP}, t)$. Send G^{pub} to the TTP.
TTP: Generate a random matrix $\mathsf{G}' \in \mathbb{F}_2^{k \times l}$ and a $(n+l)\times(n+l)$ random permutation matrix P'. Publish the systematic matrix O^{pub} generating the same $[n+l, k]$ code as

$$\left[\mathsf{G}^{\text{pub}} \middle| \mathsf{G}'\right] \mathsf{P}'.$$

decryption work like in the McEliece PKC. The difference lies in the modified public key, which ensures, that the receiver cannot decrypt all valid ciphertexts. The time complexity for Algorithm 4.2 is $\mathcal{O}(n \cdot k + n \cdot m \cdot t^2)$ operations [6]. The size of the ciphertexts is $n + l$, but as mentioned in Section 2.3, this can be reduced to $n + l - k$ by encoding the message into the error vector $\mathbf{e}$.

Note that if O^{pub} is re-used in the different instances of Algorithm 4.2, a security problem might arise when composing such instances, see discussion in Appendix A.2. For the sake of simplicity of our proofs, we henceforth assume that O^{pub} is generated each time anew.

4.1 Security Analysis

Next, we show that Algorithm 4.2 is an instance of a γ-Gap Rabin OT according to Definition 2.2.

Correctness. Observe that if parameters are chosen carefully and every party follows the protocol, Algorithm 4.2 works correctly. Let us assume

Algorithm 4.2 Transmission

Input: The security parameters m, l, t' and a k-bit message $\mathbf{m}$.

Encryption:
Obtain the receiver's public key $\mathsf{O}^{\mathrm{pub}}$.
Generate a random vector $\mathbf{e}$ of weight t' and length $2^m + l$.
Compute the ciphertext $\mathbf{c} = \mathbf{m}\mathsf{O}^{\mathrm{pub}} + \mathbf{e}$.
Send $\mathbf{c}$ to the receiver.
Following Stern's protocol with system parameters $\mathsf{O}^{\mathrm{pub}}$ and t', send a zero knowledge proof of knowledge `Proof` for the public key $\mathbf{c}$ and secret key $\mathbf{e}$ to the receiver.

Decryption:
Verify `Proof`.
Set $(\mathbf{c}_1, \mathbf{c}_2) = \mathbf{c}(\mathsf{P}')^{-1}$, where $\mathbf{c}_1$ is an n-bit vector.
Try to apply the error correction algorithm for G to $\mathbf{c}_1\mathsf{P}^{-1}$ in order to obtain $\mathbf{m}$.
if (previous step fails) **or** ($\mathbf{m}\mathsf{O}^{\mathrm{pub}} + \mathbf{c}$ has weight $\neq t'$) **then**
  return erasure.
else
  return m.

that $l = n$ and $t' = 2t+1$. Let $(\mathbf{e}_1, \mathbf{e}_2) = \mathbf{e}(\mathsf{P}')^{-1} = \mathbf{c}(\mathsf{P}')^{-1} + \mathbf{m}\left[\mathsf{G}^{\mathrm{pub}} \middle| \mathsf{G}'\right]$, where $\mathbf{e}_1$ is an n-bit vector. Then, iff $\mathbf{e}_1$ has weight $\leq t$, the decryption procedure returns the correct message $\mathbf{m}$. Else an erasure occurs or the receiver obtains a false message $\mathbf{m}' \neq \mathbf{m}$. However, the latter case is unlikely to appear, since then, the weight of $\mathbf{m}'\mathsf{O}^{\mathrm{pub}} + \mathbf{c}$ is t'. Thus, $\mathbf{m}'\mathsf{O}^{\mathrm{pub}} + \mathbf{c} + \mathbf{m}\mathsf{O}^{\mathrm{pub}} + \mathbf{c}$ has weight $\leq 2t' = 4t + 2$. It is easy to check that, for the reasonable parameters ($m > 10$ and appropriate t), the codeword $(\mathbf{m} + \mathbf{m}')\mathsf{O}^{\mathrm{pub}}$ has weight below the Gilbert-Varshamov (GV) bound for $\mathsf{O}^{\mathrm{pub}}$, which is infeasible to find, even if such a codeword exists.

Since every choice of t' below half of the GV-bound of $\mathsf{O}^{\mathrm{pub}}$ leads to a correct scheme, the parameters may be chosen, such that the probability $\mathcal{P}$ of obtaining the message $\mathbf{m}$ varies. We can compute $\mathcal{P}$ as the fraction of error vectors with no more than t entries on the positions of $\mathsf{G}^{\mathrm{pub}}$:

$$\mathcal{P} := \sum_{i=0}^{t} \frac{\binom{n}{i}\binom{l}{t'-i}}{\binom{n+l}{t'}} = 1 - \underbrace{\sum_{i=t+1}^{t'} \frac{\binom{n}{i}\binom{l}{t'-i}}{\binom{n+l}{t'}}}_{=:\mathcal{Q}^{\mathrm{H}}}. \tag{1}$$

Thus, for instance, if $n = l$ and $t' = 2t + 1$, GV bound for $\mathsf{O}^{\mathrm{pub}}$, the scheme works correctly and we have $\mathcal{P} = \mathcal{Q}^{\mathrm{H}} = 1/2$.

Gap. In fact, Algorithm 4.2 implements the γ-Gap Rabin OT for some $\gamma > 0$, since even an honest-but-curious receiver has the possibility to raise

its probability of receiving the message $\mathbf{m}$. He might choose to guess a part of the error vector or try to apply a general decoding algorithm to the erroneous word $\mathbf{c}_2$ of the code G'. These attacks are reflected in the following formula for the probability $\mathcal{Q}_A$ of an erasure for a dishonest receiver spending A operations on decryption:

$$\mathcal{Q}_A = \sum_{i=t_0}^{t_1} \frac{\binom{n}{i}\binom{l}{t'-i}}{\binom{n+l}{t'}},$$

where $t_0 > t+1$, $t_1 < t'$ (compare (1)) and the following conditions hold:

(i) Solving General Syndrome Decoding problem for $\mathbf{c}$ and $\mathsf{O}^{\mathrm{pub}}$ takes more than A operations. Note that A can be computed taking into account the (best known) attack by Canteaut and Chabaud [5] using the lower bound from [24]:

$$A \geq 2^{-t' \log_2(1-k/(n+l))}. \tag{2}$$

(ii) General Syndrome Decoding problem for $\mathbf{c}_2 = \mathbf{m}\mathsf{G}' + \mathbf{e}_2$ and G' cannot be solved in A operations if $\mathbf{e}_2$ has weight $\geq t' - t_1$.

(iii) If the weight w of $\mathbf{e}_1 = \mathbf{c}_1 + \mathbf{m}\mathsf{G}^{\mathrm{pub}}$ is larger than t_0, the receiver cannot guess a sufficiently large subset of the support of $\mathbf{e}_1$ to apply the decoding algorithm for Goppa codes. This is

$$\frac{\binom{n}{w-t}}{\binom{w}{w-t}} m^3 t^2 \geq A, \tag{3}$$

since each decoding attempt takes m^3t^2 operations [6] and there are $\binom{w}{w-t}$ correct guesses.

To the best of our knowledge there exist neither codes with better error correction ratio than binary irreducible Goppa codes nor efficient list decoding algorithm for binary irreducible Goppa codes [13]. Thus, if $\mathrm{wt}(\mathbf{e}_1) > t_0$, the receiver either has to guess part of the error or is forced to use a general decoding algorithm.

We conclude that the dishonest receiver can achieve $\mathcal{Q} = 1$ only if general decoding is easy. The gap is computed as $\gamma = \mathcal{Q}^H - \mathcal{Q}_A$. Given A, the parameters of Algorithm 4.2 must be chosen according to (2). Condition (ii) allows us to compute t_1 by substituting in (2): t' and $l+n$ with $t'-t_1$ and n, respectively. Finally, t_0 is equal to minimal w, satisfying (3). In Appendix A.1, we present the numerical computations of $\mathcal{Q}^H$ and $\mathcal{Q}_A$ for $A = \{38, 70\}$ and some proposed parameter sets.

Sender's Security. It appeared to be difficult to show a reduction to a hard problem. The general intuition behind the sender's security of

our scheme is as follows. Assume that there exists a malicious receiver algorithm R which can recover all the messages in the case of erasure. Then, R must efficiently perform either of the following tasks: 1) Correct substantially more than t errors in a $(n, k \geq n - mt)$ Goppa code; 2) Extend a Goppa code in such a way that the extended code efficiently corrects as many errors as the Goppa code of the same size.

As there is no known polynomial algorithm for either of these problems, we believe that the sender's security can be achieved in principle.

Receiver's Security.

`Proof` assures that $\mathbf{e}$ is of weight t'. In other words, the dishonest sender cannot influent $\mathcal{P}$ by playing with the error vector's weight. His ability to do it would contradict to security of Stern's ZKID.

Remark 4.1. We note that Stern's ZKID can be used for proving the validity of the McEliece encryption in the same way as it is used for the OT transmission in Algorithm 4.2. This yields a verifiable variant of McEliece encryption. Verifiable encryption has numerous applications in cryptographic protocol theory (see [4]). We leave a formal treatment of this subject for the separate paper.

Theorem 4.2. *The Sender S who can distinguish $View_S(\mathbf{m}|E = 0)$ and $View_S(\mathbf{m}|E = 1)$ can distinguish a Goppa code from a random matrix.*

Proof (Sketch).

The proof goes in two parts: First, we show that a sender S able to efficiently detect erasures (with probability 1) can distinguish the Goppa part of O^{pub} from the random part. Second, we build an oracle (in a straight forward way) which distinguishes a Goppa code from a random code using S. For simplicity of our presentation, we take $l = n$ and $t' = 2t + 1$, however our proof generalizes to almost all parameter sets.

Note, that in the case of erasure, the probability p_i that for a position i of G^{pub}, $\mathbf{e}_i = 1$ is at least $(t + 1)/n$, while for a position j of G' the probability p_j that $\mathbf{e}_j = 1$ is at most t/l. By heuristics, we can construct a distinguisher D which can tell the positions of G^{pub} apart from the ones in G'. This takes $\mathcal{O}(\text{running-time}(S) \cdot n^2)$ steps since $|p_i - p_j| \geq 1/n$. This approach only fails if $p_i \approx p_j$, which we can fix by guessing some positions of G^{pub}. See Appendix A.3 for details.

Now, given two matrices G_1 and G_2, where one is a Goppa code, we can tell, which one is the Goppa code, by querying D with $\mathsf{O}^{\text{pub}} = [\mathsf{G}_1 \mid \mathsf{G}_2]$ for the Goppa part of O^{pub}. □

4.2 Omitting Trusted Third Party

In a fully secure scheme, the key generation is performed by the receiver and its correctness is verified by the sender using Shamir's PKP ZKIP [25] (see Section 2). The basic idea is that the receiver computes the public key $\mathsf{O}^{\mathrm{pub}}$, while the sender provides its random part G' key and checks correctness by Shamir's ZKIP. The key generation protocol is summarized in Algorithm 4.3.

Algorithm 4.3 Public Key Generation Without TTP

Input: Security parameters same as in Algorithm 4.1 and $k' < 2^m - mt \in \mathbb{N}$.
Output: The public key $(\mathsf{O}^{\mathrm{pub}}, t')$.

Receiver: Generate the same public key $(\mathsf{G}^{\mathrm{pub}}, t)$ as in Algorithm 4.1.
Choose a $(n+l) \times (n+l)$ random permutation matrix P'.
Commit to the rows of $\mathsf{G}^{\mathrm{pub}}$ (one by one, k commitments in total).
Sender: Generate a random matrix $\mathsf{G}' \in \mathbb{F}_2^{k \times l}$ and send it to the receiver.
Receiver: Publish the systematic matrix $\mathsf{O}^{\mathrm{pub}}$ generating the same $[n+l, k]$ code as $\left[\mathsf{G}^{\mathrm{pub}} \middle| \mathsf{G}'\right] \mathsf{P}'$.
Sender: Choose a random subset $\mathcal{K}'$ of cardinality k' from $\{1, \ldots, k\}$.
Ask the receiver to reveal the commitments for $\mathcal{K}'$.
Compute the rows of $\left[\mathsf{G}^{\mathrm{pub}} \middle| \mathsf{G}'\right]$ with indices in $\mathcal{K}'$.
Receiver: Use Shamir's ZKIP to prove to the sender, that there is a permutation, such that the rows of $\left[\mathsf{G}^{\mathrm{pub}} \middle| \mathsf{G}'\right]$ with indices in $\mathcal{K}'$ are simultaneously in the code generated by $\mathsf{O}^{\mathrm{pub}}$.

The last step of Algorithm 4.3 requires some additional explanation: After the second last step, the sender knows a $k' \times n$ submatrix K of $\left[\mathsf{G}^{\mathrm{pub}} \middle| \mathsf{G}'\right]$ and can compute the $n + l - k$ dimensional kernel of $\mathsf{O}^{\mathrm{pub}}$ given by a matrix H. Now we can take H and k' as system parameters for Shamir's Permuted Kernel ZKIP. If the receiver is honest and has followed the protocol, he knows the secret permutation $\Pi = \mathsf{P}'$ corresponding to the user's identity K, i.e. a permutation such that $\mathsf{K} \cdot \Pi \cdot \mathsf{H} = \mathbf{0}$. Thus, the honest receiver can employ Shamir's ZKIP to convince the sender by a zero-knowledge proof that he followed the protocol, while the dishonest receiver will be revealed.

Note that G' can be generated using a pseudorandom generator. In this case, only a seed to the generator needs to be transmitted, hereby communication cost can be reduced.

5 1-out-of-2 String Oblivious Transfer

As a generalization of the previous scheme, we can construct a substantially more efficient protocol for 1-out-of-2 String OT. Unfortunately, this construction inherits the drawback of the previous scheme – we are unable to formally prove its sender's security. In fact, the sender's security proof would be a generalization of that of our Rabin OT scheme.

In this section, we briefly sketch this 1-out-of-2 String OT scheme and its security intuition, without giving formal proofs.

We assume the players' inputs to the protocol to be random since there exists a reduction by Beaver [2] which allows to convert such randomized OT into OT with actual inputs. Let the following be the security parameters: the matrix $\mathsf{Q} \in \mathbb{F}_2^{k \times n}$ chosen uniformly at random, and $\mathsf{G} \in \mathbb{F}_2^{k \times n}$, as before, be a canonical generator matrix of an $[n, k \geq n - mt, 2t + 1]$ binary irreducible Goppa code. Encryption is done using a variant of the McEliece PKC, we assume that the corresponding inputs' length is whatever prescribed by the encryption algorithm, denoted a bits for certainty.

Our 1-out-of-2 String OT protocols is summarized in Algorithm 5.1.

Algorithm 5.1 1-out-of-2 String OT

Input: Security parameters: $m, k, k', t, \mathsf{G}, \mathsf{Q}$
Sender: $\mathbf{b}_0, \mathbf{b}_1 \in \mathbb{F}_2^a$; **Receiver:** $c \in \{0, 1\}$
Output: Sender: none; **Receiver:** b_c

Receiver: Generate random permutation matrices $\mathsf{P}', \mathsf{P}'' \in \mathbb{F}_2^{n \times n}$ and a random matrix of rank k': $\mathsf{S}' \in \mathbb{F}_2^{k' \times k}$. Set $\mathsf{C}_c = \mathsf{S}'\mathsf{G}\mathsf{P}'$, $\mathsf{C}_{1-c} = \mathsf{S}'\mathsf{Q}\mathsf{P}''$.
Send $[\mathsf{C}_0|\mathsf{C}_1]$ to the receiver.
Prove using Shamir's ZKIP that $[\mathsf{C}_0|\mathsf{C}_1]$ is a permuted subcode of $[\mathsf{G}|\mathsf{Q}]$.
Sender: Reject if the proof fails, otherwise
For $i = 0, 1$: Encrypt $\mathbf{b}_i$ as follows: $\mathbf{b}_i\mathsf{C}_i + \mathbf{e}_i$, where $\mathbf{e}_i$ is random vector of weight t, and send the encryption.
Receiver: Decrypt and output $\mathbf{b}_c$.

Note that the communication cost of this protocol can be substantially reduced, if only the non-systematic parts of the codes are dealt with. This modification will also require using the IND-CCA2 conversion for encryption [17]. Also, Q can be computed using a pseudorandom generator such that only its seed will be obtained by coin flipping.

Intuition. We provide only a sketch of the security analysis.

Correctness. If both players follow the protocol, then the receiver is not rejected by the sender and is able to decode C_c which is a subcode of G. Hence, the decoding algorithm of G can be used.

Sender's Security. Assume that the receiver is honest-but-curious, i.e., he tries to compute both inputs, but still follows the protocol. In order to compute $\mathbf{b}_{1-c}$, he must decode a subcode of the random code, i.e., solve Problem *(ii)* of Definition 2.3.

Now, suppose that the IND-CPA version of the McEliece PKC [21] is used for encryption, then the input $\mathbf{b}_{1-c}$ which is encrypted on the subcode of Q is indistinguishable from random according [21, Lemma 3].

If we use only non-systematic parts of the codes, then we must employ the IND-CCA2 conversion [17] since the "message" part of the encryption will be sent in open in this case. Indistinguishability of one of the inputs will follow in a way similar to the variant above. Note, that the price to pay here is the additional assumption: the random oracle model, which is not required by the variant above.

Now, assume that the receiver is fully malicious. The ZK proof will convince the sender that $[\mathsf{C}_c|\mathsf{C}_{1-c}]$ is indeed a permutation of $[\mathsf{G}|\mathsf{Q}]$. Unfortunately, the receiver may distribute the columns of G and Q into both C_c and C_{1-c}. Then, the security proof will boil down to proving the receiver's inability to efficiently decode an extended Goppa code. In fact, we will need a generalization of the security proof for our Rabin OT protocol: here, the receiver does not necessarily distributes G and Q as "half-to-half" in the subcodes. As it was mentioned in the previous section, such the proof is not easy to construct.

Receiver's Security. The malicious sender who learns the receiver's choice must either distinguish the subcode of G from that of Q hence solving Problem *(i)* or recover the permutations $\mathsf{P}', \mathsf{P}''$ solving Problem *(iv)* of Definition 2.3.

Employing Cut-And-Choose. We note, although do not prove it formally, that in the above algorithm, one can use the cut-and-choose technique instead of the zero-knowledge proof in order for the sender to check that the keys were formed correctly. This would remove the above mentioned problem with the sender's security proof. However, in this case, our protocol would become quite similar to that of [9], since it would use the same machinery to reduce the advantage of malicious receiver.

Acknowledgments. The authors would like to thank anonymous reviewers from Eurocrypt 2008. The second author would also like to thank Yang Cui for his pointer on verifiable encryption.

References

1. W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2001.
2. D. Beaver. Precomputing oblivious transfer. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1995.
3. M. Bellare and S. Micali. Non-interactive oblivious transfer and spplications. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 547–557. Springer, 1989.
4. Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.
5. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEETIT: IEEE Transactions on Information Theory*, 44, 1998.
6. N. Courtois, M. Finiasz, and N.Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248, pages 157–174. Springer-Verlag, 2001.
7. Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 350–354. Springer, 1987.
8. I. Damgård and J. Nielsen. Commitment schemes and zero-knowledge protocols. Lecture notes, University of Aarhus, February 2008. Available at: http://www.daimi.au.dk/~ivan/ComZK08.pdf.
9. R. Dowsley, J. van de Graaf, J. Müller-Quade, and A. Nascimento. Oblivious transfer based on the McEliece assumptions. Accepted to ICITS 2008, August 2008.
10. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.
11. O. Goldreich. *Foundations of Cryptography - Volume 2 (Basic Applications)*. Cambridge University Press, 2004.
12. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM, 1987.
13. V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.
14. I. Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 394–409. Springer, 2004.
15. Y. Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 78–95. Springer, 2005.
16. J. Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31. ACM, 1988.
17. K. Kobara and H. Imai. Semantically secure McEliece public-key cryptosystems - conversions for McEliece PKC. In *Practice and Theory in Public Key Cryptography - PKC '01 Proceedings*. Springer Verlag, 2001.

18. F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correctiong Codes*. North-Holland Amsterdam, 7 edition, 1992.
19. R.J. McEliece. A public key cryptosystem based on algebraic coding theory. *DSN progress report*, 42-44:114–116, 1978.
20. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001.
21. Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the McEliece cryptosystem without random oracles. Accepted to Special Issue of Designs, Codes and Cryptography, 2008. Available at: http://www.springerlink.com/content/382002h225822816/.
22. G. Poupard. A realistic security analysis of identification schemes based on combinatorial problems. *European Transactions on Telecommuncations*, 8(5):417–480, September/October 1997.
23. M.O. Rabin. How to exchange secrets by oblivious transfer. Technical report, Aiken Computation Laboratory, Harvard University, 1981. Tech. Memo TR-81.
24. N. Sendrier. On the security of the McEliece public-key cryptosystem. In M. Blaum, P.G. Farrell, and H. van Tilborg, editors, *Proceedings of Workshop honoring Prof. Bob McEliece on his 60th birthday*, pages 141–163. Kluwer, 2002.
25. A. Shamir. An efficient identification scheme based on permuted kernels. In *Proc. of Crypto'89*, volume 435 of *LNCS*, pages 606–609. Springer Verlag, 1990.
26. J. Stern. A new identification scheme based on syndrome decoding. In *Advances in Cryptology - CRYPTO'93*, volume 773 of *LNCS*. Springer Verlag, 1994.
27. Serge Vaudenay. Cryptanalysis of the Chor–Rivest cryptosystem. *J. Cryptology*, 14(2):87–100, 2001.
28. S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

Appendix A Details on Security of Rabin OT

A.1 Examples of Security Parameters

We can achieve reasonable values for $\mathcal{P}$ and $\mathcal{Q}_{2^{80}}$, compare Table A.1. In fact, one can even use the dishonest receiver's strategy in a positive way, i.e., to raise the chance of obtaining the message. The work factor for decryption is then given by Equation (3). This is useful for protocols like in [7], where we need to ensure that the receiver gets at least half of the messages, compare Section 3.

Parameters				Size Public Key	Size Ciphertext	Decryption		$\mathcal{Q}_A$	
m	t	t'	l	$= k(n+l-k)$	$= n+l-k$	runtime	$\mathcal{Q}^{\mathrm{H}}$	$A = 2^{38}$	$A = 2^{80}$
12	200	$2t+1$	2^{12}	1,377 KBytes	812 Bytes	2^{26}	0.5	0.41	0.11
13	402	$2t+13$	2^{13}	4,974 KBytes	1,677 Bytes	$2^{28.5}$	0.66	0.61	0.36
14	800	$2t+1$	2^{14}	17,874 KBytes	3,448 Bytes	2^{31}	0.5	0.46	0.29

Table A.1. Parameter sets for the Rabin OT

Example 1. With the first parameter set from Table A.1 and a receiver spending up to 2^{35} operations on each decryption, we can obtain a 1-out-of-2 OT by Crépeau's construction, which fails with probability less than 2^{-30} if we choose $N = 180$.

A.2 Reaction Attack

Note that if the same public key $\mathsf{O}^{\mathrm{pub}}$ is used in the different instances of Algorithm 4.2, the dishonest sender can adaptively influent the erasure probability, as long as he gets feedback whether an erasure occurred or not. Suppose that an attacker learns that the receiver cannot decode a certain ciphertext. Then, the sender can choose to modify the corresponding error vector only slightly for the next encryption. Thus, by statistics, the sender could identify the columns of $\mathsf{G}^{\mathrm{pub}}$ in $\mathsf{O}^{\mathrm{pub}}$, which breaks the security of our scheme. Nevertheless, the sender should be cautious, as the receiver might detect such manipulations by comparing ciphertexts.

This might get important as feedback may well come from the higher level protocols (like Crépeau's protocol), for which oblivious transfer is used as a primitive. However, there are plenty of possible countermeasures against an attack by feedback. For instance, when the conversion [17] is used for encryption, the task of tuning the erasure probability is not at all trivial.

A.3 Proof Details for Receiver's Security

The distinguisher D works as follows. Repeat the following until $n^{2+\epsilon}$ $(0 < \epsilon < 1)$ erasure views are encountered:

- Generate a view of the sender using some error vector $\mathbf{e}$ (distinct each time) and submit it to S
- Each time S outputs "erasure", remember on which columns the error locations of $\mathbf{e}$ were.

Note that the expected running time will be $n^{2+\epsilon}/\mathcal{Q}$, where $\mathcal{Q}$ is the erasure probability. Hence, D is efficient.

Then, consider the "score" of each column. For those of $\mathsf{G}^{\mathrm{pub}}$, the expected score is at least $(t+1)n^{1+\epsilon}$, since at least $t+1$ errors are needed to cause an erasure. Hence, for the columns of G', the expected score is at most $(t'-t-1)n^{2+\epsilon}/l$ as at most $t'-t-1$ errors are left for G'. Now, the standard Chernoff bound can be applied in order to show that one can distinguish between two random variables with the above expectations with negligible (in n) probability of error.

It easy to check that the above reasoning works when $(t+1)/n > (t'-t-1)/l$. The only case, when it breaks down is when the above expectations are too close to each other such that the Chernoff bound does not apply. However, this can be fixed in the following way: guess a few positions of G^{pub} and run D. Note that once $t+1$ candidate columns for G^{pub} are obtained, they can be easily verified by placing the error locations on them and submitting such the view to S. If the initial guess was wrong, guess a different set of columns and start over. Since the probability of the correct guess is non-negligible, the expected running time is polynomial in n.

In case of $(t'-t-1)/l < (t+1)/n$, we slightly modify the construction of D: it will iterate until $n^{2+\epsilon}$ *non-erasure* views are encountered. Note that in this case, the expected score of G^{pub}'s column is at most $tn^{1+\epsilon}$, while that of G' is at least $(t'-t)n^{2+\epsilon}/l$. Thus, the reasoning before applies in a similar way.