J. Math. Cryptol. **5** (2011), 277–301
DOI 10.1515/jmc-2012-0001

On the Bringer–Chabanne EPIR protocol for polynomial evaluation

Yeow Meng Chee, Huaxiong Wang and Liang Feng Zhang

Communicated by Spyros Magliveras

Abstract. Extended private information retrieval (EPIR) was defined by Bringer, Chabanne, Pointcheval and Tang at CANS 2007 and generalized by Bringer and Chabanne at AFRICACRYPT 2009. In the generalized setting, EPIR allows a user to evaluate a function on a database block such that the database can learn neither which function has been evaluated nor on which block the function has been evaluated and the user learns no more information on the database blocks except for the expected result. An EPIR protocol for evaluating polynomials over a finite field L was proposed by Bringer and Chabanne in [Lecture Notes in Comput. Sci. 5580, Springer (2009), 305–322]. We show that the protocol does not satisfy the correctness requirement as they have claimed. In particular, we show that it does not give the user the expected result with large probability if one of the coefficients of the polynomial to be evaluated is primitive in L and the others belong to the prime subfield of L.

Keywords. Extended private information retrieval, correctness.

2010 Mathematics Subject Classification. 94A60.

1 Introduction

Extended private information retrieval (EPIR) was motivated by privacy-preserving biometric authentication and formally defined in [6]. It enables a user to privately evaluate a fixed and public function with two inputs, one chosen block from a database and one additional string. Two EPIR protocols were proposed in [6]. One is for testing equality and the other is for computing weighted Hamming distance. As a cryptographic primitive, EPIR has been generalized by [5] in order to attain more flexibility. In the generalized setting, the function to be evaluated is neither fixed nor public. Instead, it is chosen from a set of public functions by the user. A new EPIR protocol in the generalized setting was proposed in [5]. As

The research is supported in part by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03.

noted in [6], EPIR is indeed a combination of private information retrieval [12] and general secure two-party computation [18].

Related Work. Private information retrieval (PIR) was introduced by Chor, Goldreich, Kushilevitz and Sudan in [12]. It allows a user to retrieve a data item from a database such that the database cannot learn which item the user is interested in. The requirement on the privacy of the identity of the retrieved data item is called *user privacy*. The main measure of the efficiency of a PIR protocol is its *communication complexity*, i.e., the total number of bits exchanged by the user and the database for retrieving a *single bit*. PIR protocols have been constructed in both the *information-theoretic* setting [1–3, 10, 12, 13, 21, 29, 31] and the *computational* setting [7, 9, 11, 16, 20, 23, 24, 26, 30]. In an information-theoretic PIR protocol, the database learns *absolutely* no information on which item the user is interested in, even if it has unlimited computing power. On the other hand, in a computational PIR (CPIR) protocol, the identity of the retrieved data item is not revealed only if the database is polynomial-time and cannot efficiently solve certain number-theoretic problems, i.e., certain cryptographic assumptions hold. For example, the PIR protocol of [11] is a *two-database* CPIR protocol in which each database cannot figure out which item the user is interested in under the assumption that one way functions exist. EPIR protocols of [5, 6] are mostly close to the *single-database* CPIR protocols. The first single-database CPIR protocol was proposed by Kushilevitz and Ostrovsky in [23]. It achieves the user privacy under the assumption that deciding quadratic residuosity is hard and has communication complexity $O(N^c)$ for any small constant $c > 0$, where N is the size of the database. Subsequently, Cachin, Micali and Stadler [7] constructed a single-database CPIR protocol of communication complexity $O(\log^8(N))$ under the Φ-hiding assumption. So far, the most efficient single-database CPIR protocol was obtained by Gentry and Ramzan [16] under the assumption that the decision subgroup problem is hard. It requires the user to exchange $O(k + d)$ bits with the database for retrieving d bits, where $k \geq \log N$ is the security parameter. Other constructions of single-database CPIR protocols can be found in [9, 20, 24, 30].

PIR does not provide any privacy for the database. Typically, the user may obtain a large number of data items in an execution of a PIR protocol. In order to prevent the user from obtaining more than one data item in any execution of a PIR protocol, Gertner et al. [17] introduced the notion of *data privacy* and proposed transformations from information-theoretic PIR protocols to the so-called symmetrically private information retrieval (SPIR) protocols which meet the data privacy. The SPIR protocols of [17] are in the information-theoretic setting. SPIR can be defined in the computational setting as well. Following the security definition of general secure two-party and multi-party computation [18],

in the computational setting, a PIR protocol is said to achieve data privacy if, for any query, the user cannot tell whether it is interacting with a real-database which has N data items or a simulator which only knows the retrieved data item. Interestingly, single-database SPIR protocols in the computational setting are essentially communication-efficient 1-out-of-N oblivious transfer (OT) protocols [4,14,19,22,28]. Oblivious transfer [28] is a fundamental cryptographic primitive, on which any secure two-party and multi-party computation can be built [22] in an unconditionally secure way. A 1-out-of-N OT allows a receiver Bob to choose one of the N secrets held by a sender Alice such that Alice learns no information on Bob's choice and Bob cannot learn more except the secret he chooses. Naor and Pinkas [27] proposed transformations from any PIR protocols to SPIR protocols in the computational setting. Their transformation requires only one execution of a given PIR protocol and $\log N$ executions of a 1-out-of-2 OT protocol. The notion of EPIR [5,6] is essentially a generalization of SPIR in the computational setting.

EPIR is also related to selective private function evaluation [8], oblivious polynomial evaluation [27] and private keyword search [15]. A selective private function evaluation protocol [8] allows a client to privately evaluate a public function on the inputs held by one or more servers. Comparing with EPIR, the client only decides on which inputs the public function will be evaluated. An oblivious polynomial evaluation protocol [27] allows a receiver to privately evaluate a polynomial function on his input, where the polynomial is held by a sender. Comparing with EPIR, the function to be evaluated is not known to the receiver and the input on which the function is evaluated is not known to the sender. A private keyword search protocol [15] allows a client to privately search a database with a keyword such that he learns the associated record if the keyword is contained in the database and learns nothing otherwise. In a sense, EPIR can also be seen as a generalization of the above problems.

Results. The protocol described in [5, Section 4.3] will be our main topic in this paper and termed as Bringer–Chabanne EPIR protocol from now on. It was claimed [5] that the protocol enables a user to privately evaluate any polynomial $F(t) \in L[t]$ on a chosen database block R_i, where $L = \mathrm{GF}(p^n)$ is the field extension of degree n of the prime field $K = \mathrm{GF}(p)$. We study the correctness of the Bringer–Chabanne EPIR protocol and show that it may fail frequently.

In particular, we show that, by executing the protocol, the user with input $(F(t), i) \in L[t] \times [N]$ does not learn the expected result (i.e., $F(R_i)$) with a large probability if $F(t) \in \mathcal{P}$, where $\mathcal{P}$ is defined as follows:

$$\mathcal{P} = \Big\{ f(t) = \sum_{k=0}^{d} f_k t^k : \exists\, 0 \le l \le d \text{ such that } f_l \in L \text{ is of order } p^n - 1 \text{ and } f_k \in K \text{ for every } k \ne l \Big\}.$$

Methodology. Our argument is by contradiction. To simplify the argument, we first give a restricted version of the Bringer–Chabanne EPIR protocol. In the restricted version, the database is deterministic and has only one block, i.e., $N = 1$. We note that if the Bringer–Chabanne EPIR protocol satisfies the correctness requirement, then so does the restricted version. We then show that the restricted version does not satisfy the correctness requirement if the polynomial to be evaluated is in $\mathcal{P}$. This result allows us to conclude that the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement as [5] has claimed.

Organization. The remainder of this paper is organized as follows. In Section 2, we recall the definition and security model of EPIR [5]. In Section 3, we recall the Bringer–Chabanne EPIR protocol. In Section 4, we give a restricted version of the Bringer–Chabanne EPIR protocol and show that the restricted version fails frequently if the polynomial to be evaluated is in $\mathcal{P}$. Section 5 summarizes the results of the paper.

2 Preliminaries

2.1 Definition

Following the definition of [5], a single-database EPIR protocol is a protocol between a database $\mathcal{DB}$ who has N blocks $(R_1, \ldots, R_N) \in (\{0,1\}^{l_1})^N$ and a user $\mathcal{U}$ who wants to evaluate $F(R_i)$ for a function $F \in \mathcal{F}$ and an index $i \in [N]$, where $\mathcal{F}$ is a set of functions from $\{0,1\}^{l_1}$ to $\{0,1\}^*$ and public. Such a protocol allows $\mathcal{U}$ to learn $F(R_i)$ but no more information on the database blocks while $\mathcal{DB}$ learns no information on (F, i).

The above definition of EPIR is a generalization of [6] and provides the user with more flexibility of choosing the function F from a large set $\mathcal{F}$. In the context of this definition, the EPIR for testing equality [6] has

$$\mathcal{F} = \big\{ \text{IsEqual}(\cdot, X) : X \in \{0,1\}^{l_1} \big\},$$

where

$$\text{IsEqual}(R_i, X) = \begin{cases} 1 & \text{if } R_i = X, \\ 0 & \text{otherwise.} \end{cases}$$

The EPIR for computing weighted Hamming distance [6] has

$$\mathcal{F} = \left\{ d_w(\cdot, X) : X \in \{0,1\}^{l_1}, w \in \mathbb{N}^{l_1} \right\},$$

where

$$d_w(R_i, X) = \sum_{j=1}^{l_1} w_j \cdot (R_i^{(j)} \oplus X^{(j)}).$$

(For every $j \in [l_1]$, $R_i^{(j)}$ and $X^{(j)}$ are the j-th bits of R_i and X, respectively.)

2.2 Security model

As in [5, 6], we denote by $\mathsf{retrieve}(F, i)$ the query made by a user with input $(F, i) \in \mathcal{F} \times [N]$. Without further notice, algorithms are assumed to be polynomial-time.

If an algorithm $\mathcal{A}$ runs in k stages, then we shall write $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k)$. The security is evaluated by an experiment between an attacker and a challenger, where the challenger simulates the protocol executions and answers the attacker's oracle queries. For $\mathcal{A}$ a probabilistic algorithm, we denote by $\mathcal{A}(\mathcal{O}, \mathsf{retrieve})$ the action to run $\mathcal{A}$ with access to any polynomial number of $\mathsf{retrieve}$ queries generated or answered (depending on the position of the attacker) by the oracle $\mathcal{O}$. A function $\tau : \mathbb{Z} \to \mathbb{R}$ is said to be *negligible* if for any polynomial P, there is an integer N_P such that $\tau(n) \leq 1/P(n)$ for every $n \geq N_P$. If $\tau(n)$ is negligible, then $1 - \tau(n)$ is said to be *overwhelming*.

Correctness. An EPIR protocol is said to be *correct* if any query $\mathsf{retrieve}(F, i)$ returns the correct value of $F(R_i)$ with an overwhelming probability when $\mathcal{U}$ and $\mathcal{DB}$ follow the protocol specification.

User Privacy. Informally, an EPIR protocol is said to *respect user privacy* if for any query $\mathsf{retrieve}(F, i)$, $\mathcal{DB}$ learns no information on (F, i). Formally, an EPIR protocol is said to respect user privacy if any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$, acting as a malicious database, has only a negligible advantage $|\Pr[b' = b] - \frac{1}{2}|$ in the following experiment:

$\mathsf{Exp}_{\mathcal{A}}^{\text{user-privacy}}$

$$\begin{array}{rcl}
(R_1, \ldots, R_N) & \leftarrow & \mathcal{A}_1(1^l) \\
1 \leq i_0, i_1 \leq N; F_0, F_1 \in \mathcal{F} & \leftarrow & \mathcal{A}_2(\text{Challenger}; \mathsf{retrieve}) \\
b & \leftarrow & \{0,1\} \\
\emptyset & \leftarrow & \mathcal{A}_3(\text{Challenger}; \mathsf{retrieve}(F_b, i_b)) \\
b' & \leftarrow & \mathcal{A}_4(\text{Challenger}; \mathsf{retrieve})
\end{array}$$

Database Privacy. Informally, an EPIR protocol is said to *respect database privacy* if a malicious user $\mathcal{U}$ cannot learn more information than $F'(R_{i'})$ for some $(F', i') \in \mathcal{F} \times [N]$ via a query retrieve. This intuitive description can be formalized via simulation principle by saying that the user $\mathcal{U}$ cannot determine whether he is interacting with a simulator which takes only $(i', F'(R_{i'}))$ as input, or with $\mathcal{DB}$. We denote by $\mathcal{S}_0$ the database $\mathcal{DB}$. Formally, an EPIR protocol is said to respect database privacy if there is a simulator $\mathcal{S}_1$, which receives an auxiliary input $(i', F'(R_{i'}))$ from a *hypothetical oracle* $\mathcal{O}$ for every query retrieve, such that any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, acting as a malicious user, has only a negligible advantage $|\Pr[b' = b] - \frac{1}{2}|$ in the following experiment:

$\mathsf{Exp}_{\mathcal{A}}^{\text{database-privacy}}$

$$\left|\begin{array}{rcl} b & \leftarrow & \{0,1\} \\ (R_1, \ldots, R_N) & \leftarrow & \mathcal{A}_1(1^l) \\ b' & \leftarrow & \mathcal{A}_2(\mathcal{S}_b; \mathsf{retrieve}) \end{array}\right.$$

We remark that the hypothetical oracle $\mathcal{O}$ is assumed to have unlimited computing resources, and $\mathcal{S}_1$ always learns exactly the input related to the request made by the attacker.

3 Bringer–Chabanne EPIR protocol

The EPIR protocols for testing equality and computing weighted Hamming distance of [6] are based on a pre-processing technique. Specifically, the user sends an encryption of its input (F, i) to $\mathcal{DB}$, who then computes a temporary database which contains an encryption of $F(R_i)$. Finally, the user executes a single-database CPIR protocol with $\mathcal{DB}$ to retrieve the encryption of $F(R_i)$. This technique does not allow the evaluation of generic functions and incurs heavy computation during the computation of the temporary database. The Bringer–Chabanne EPIR protocol aims to avoid these deficiencies. It is based on ElGamal encryption schemes over the multiplicative groups of finite fields.

3.1 ElGamal encryption scheme

Let p be a prime and $K = \mathrm{GF}(p)$ be the finite field of order p. Let $L = \mathrm{GF}(p^n)$ be the finite field of order p^n and $\mathbb{G} = L^\times$ be its multiplicative group of order $q = p^n - 1$ for an integer $n \geq 2$. Let g be a generator of $\mathbb{G}$. The ElGamal encryption scheme over $\mathbb{G}$ is a triplet of algorithms $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with Gen, Enc, Dec as follows:

(i) Gen is a key generation algorithm which takes as input a security parameter 1^k and proceeds as follows:

- generates the parameters p, n, q and g;
- picks $x \leftarrow \mathbb{Z}_q$ and computes $y = g^x$;
- outputs $pk = (q, g, y)$ as the public key and $sk = x$ as the secret key.

(ii) Enc is an encryption algorithm which takes as input a plaintext $m \in \mathbb{G}$, picks $r \leftarrow \mathbb{Z}_q$ and outputs $c = (g^r, y^r m)$ as the ciphertext.

(iii) Dec is a decryption algorithm which takes as input a ciphertext $c = (c_1, c_2) \in \mathbb{G}^2$ and outputs $c_2 \cdot c_1^{-x}$.

3.2 Requirements on database blocks and functions

Following the notation in Section 3.1, let $\alpha \in L$ be a primitive element of the field extension L/K. Then there is a polynomial $G(t) \in K[t]$ of degree $< n$ such that $G(\alpha) = g$. Let $x \in \mathbb{Z}_q$ and $Y(t) \in K[t]$ be the polynomial of degree $< n$ such that $Y(\alpha) = y = g^x$.

For the Bringer–Chabanne EPIR protocol to be correct, it is required in [5] that for every $j \in [N]$, the database block R_j should belong to $\mathbb{D}$, where

$$\mathbb{D} = \{\beta \in \mathbb{G} : Y(\beta) = G(\beta)^x \text{ and } G(\beta) \neq 0\}.$$

The function to be evaluated by $\mathcal{U}$ can be any polynomial over L, i.e., $\mathcal{F} = L[t]$.

3.3 Bringer–Chabanne EPIR protocol

Figure 1 is the Bringer–Chabanne EPIR protocol, where most notation is adopted from Sections 3.1 and 3.2. The authors of the protocol expect to embed the description of the polynomial $F(t) \in L[t]$ chosen by $\mathcal{U}$ into an ElGamal ciphertext such that it can be evaluated by $\mathcal{DB}$ in an oblivious way.

The correctness of the Bringer–Chabanne EPIR protocol was claimed in [5] as follows.

Claim 3.1 ([5, Section 4.4]). *A query (say* retrieve(F, i)*) gives the expected result (i.e., $F(R_i)$) as soon as there is no index j for which one of the values $G(R_j)$ or $Y(R_j)$ is zero, which may occur only with a negligible probability in practice, leading to the correctness of the EPIR protocol.*

(i) $\mathcal{U}$: Generates an ElGamal key pair (pk, sk), where $pk = (q, g, y)$, $y = g^x$, and $sk = x$ is randomly chosen from $\mathbb{Z}_q$. $\mathcal{U}$ also sends pk to let $\mathcal{DB}$ the possibility to verify the validity of pk as an ElGamal public key. In practice, the validity of pk can be certified by a TTP, and the same pk can be used by the user for all his queries.

(ii) $\mathcal{U}$: For any polynomial function $F : \mathrm{GF}(p^n) \to \mathrm{GF}(p^n)$ and any index $1 \le i \le N$, computes $C_1, \ldots, C_N$ and sends them to $\mathcal{DB}$ where

- $C_i = \mathsf{Enc}(F(\alpha) + r) = (G(\alpha)^{r_i}, Y(\alpha)^{r_i}(F(\alpha) + r))$, and
- $C_j = \mathsf{Enc}(1) = (G(\alpha)^{r_j}, Y(\alpha)^{r_j})$ for all $j \ne i$,

with randomly chosen $r \in \mathrm{GF}(p)$, $r_j \in \mathbb{Z}_q$ $(1 \le j \le N)$. Each C_j can be written as $C_j = (V_j(\alpha), W_j(\alpha))$ where V_j and W_j are polynomial over $\mathrm{GF}(p)$ of degree at most $n - 1$.

(iii) $\mathcal{DB}$: After reception of the C_j, checks that they are nontrivial ElGamal ciphertexts and computes $C_j(R_j) = (V_j(R_j), W_j(R_j))$ by replacing each occurrence of α (resp. α^l for all power $l < n$) with R_j (resp. with R_j^l).

(iv) $\mathcal{DB}$: Performs the product of all the C_j together with a random encryption of 1, say $\mathsf{Enc}(1) = (g^{r'}, y^{r'})$, sends $\mathsf{Enc}(1) \times \prod_{j=1}^N C_j(R_j) = \big(g^{r'} \prod_{j=1}^N G(R_j)^{r_j}, y^{r'} \big(\prod_{j=1}^N Y(R_j)^{r_j}\big)(F(R_i) + r)\big)$ to $\mathcal{U}$.

(v) $\mathcal{U}$: Outputs $\mathsf{Dec}(sk, \mathsf{Enc}(1) \prod_{j=1}^N C_j(R_j)) - r$ as $F(R_i)$.

Figure 1. Bringer–Chabanne EPIR protocol.

4 On the incorrectness of the Bringer–Chabanne EPIR protocol

In this section, we show that the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement defined in Section 2.2. To simplify the argument, we give a restricted version of the Bringer–Chabanne EPIR protocol in which $\mathcal{DB}$ is deterministic and $N = 1$. The restricted version satisfies the correctness requirement as long as the Bringer–Chabanne EPIR protocol satisfies the correctness requirement. Then we turn to study the incorrectness of the restricted version.

4.1 Restricted version

At step (iv) of the Bringer–Chabanne EPIR protocol, $\mathcal{DB}$ is randomizing the product $\prod_{j=1}^N C_j(R_j)$ and sending $\mathsf{Enc}(1) \cdot \prod_{j=1}^N C_j(R_j)$ to the user. We note that the user could have computed the same output if $\mathcal{DB}$ merely sends $\prod_{j=1}^N C_j(R_j)$.

(i) $\mathcal{U}$: Generates an ElGamal key pair (pk, sk), where $pk = (q, g, y)$, $y = g^x$, and $sk = x$ is randomly chosen from $\mathbb{Z}_q$. $\mathcal{U}$ also sends pk to let $\mathcal{DB}$ the possibility to verify the validity of pk as an ElGamal public key. In practice, the validity of pk can be certified by a TTP, and the same pk can be used by the user for all his queries.

(ii) $\mathcal{U}$: For any polynomial function $F : \mathrm{GF}(p^n) \to \mathrm{GF}(p^n)$, computes $C = \mathsf{Enc}(F(\alpha) + r) = (G(\alpha)^s, Y(\alpha)^s(F(\alpha) + r))$ and sends it to $\mathcal{DB}$ where $r \in \mathrm{GF}(p)$, $s \in \mathbb{Z}_q$ are randomly chosen. The ciphertext C can be written as $C = (V(\alpha), W(\alpha))$ where V and W are polynomials over $\mathrm{GF}(p)$ of degree at most $n - 1$.

(iii) $\mathcal{DB}$: After reception of C, checks that it is a nontrivial ElGamal ciphertext and computes $C(R) = (V(R), W(R))$ by replacing each occurrence of α (resp. α^l for all power $l < n$) with R (resp. with R^l).

(iv) $\mathcal{DB}$: Sends $C(R)$ to $\mathcal{U}$.

(v) $\mathcal{U}$: Outputs $\mathsf{Dec}(sk, C(R)) - r$ as $F(R)$.

Figure 2. A restricted version of the Bringer–Chabanne EPIR protocol.

Therefore, we can safely modify step (iv) of the Bringer–Chabanne EPIR protocol such that $\mathcal{DB}$ merely sends $\prod_{j=1}^{N} C_j(R_j)$ to $\mathcal{U}$ with no impact on the correctness of the protocol. Let $i = N = 1$. Then we have the restricted version (see Figure 2).

Clearly, if Claim 3.1 holds, then we have:

Claim 4.1. *A query (say* $\mathsf{retrieve}(F, 1)$*) in an execution of the restricted version gives* $\mathcal{U}$ *the expected result (i.e.,* $F(R)$*) for any* $R \in \mathbb{G}$ *satisfying* $Y(R) = G(R)^x$ *and* $G(R) \neq 0$*.*

4.2 Counterexample

By a counterexample we show that Claim 4.1 does not hold. Let $p = 2$, $n = 3$, $K = \mathrm{GF}(2)$, $L = \mathrm{GF}(2^3)$ and $\mathbb{G} = L^\times$. Let $\alpha = g \in \mathbb{G}$ be a generator of $\mathbb{G}$ with minimal polynomial $\mathrm{Min}_g(t) = t^3 + t + 1 \in K[t]$. Figure 3 is an execution of the restricted version which does not give $\mathcal{U}$ the expected result.

(i) $\mathcal{U}$: Picks a private key $sk = x = 6 \in \mathbb{Z}_7$, sets $y = g^2 + 1$ and $pk = (7, g, y)$. (pk, sk) is a pair of public and private keys for the ElGamal encryption scheme over group $\mathbb{G}$. $\mathcal{U}$ sends pk to $\mathcal{DB}$ such that $\mathcal{DB}$ can verify the validity of pk as an ElGamal public key. Clearly, $g = G(\alpha)$ and $y = Y(\alpha)$ for polynomials $G(t) = t$, $Y(t) = t^2 + 1 \in K[t]$ of degree less than 3. The field elements $R \in L$ which satisfy equality $Y(R) = G(R)^x$ are g, g^2 and $g^2 + g$.

(ii) $\mathcal{U}$: For a polynomial function $F(t) = g \in L[t]$, takes $s = 6 \in \mathbb{Z}_7$, $r = 1 \in K$ and computes the ciphertext $C = \mathsf{Enc}(F(\alpha) + r) = (G(\alpha)^s, Y(\alpha)^s(F(\alpha)+r)) = (g^6, (g^2+1)^6(g+1)) = (g^2+1, g^2+g)$ and sends it to $\mathcal{DB}$. Clearly, we have that $V(t) = t^2 + 1$ and $W(t) = t^2 + t$.

(iii) $\mathcal{DB}$: Sets the database block to be $R = g^2 + g \in \mathbb{G}$. After receiving the ciphertext $C = (g^2 + 1, g^2 + g)$ from $\mathcal{U}$, $\mathcal{DB}$ checks that C is a nontrivial ElGamal ciphertext and computes $C(R) = (V(R), W(R)) = (R^2 + 1, R^2 + R) = (g + 1, g^2)$ by replacing each occurrence of α (resp. α^l for all power $l < n$) with R (resp. with R^l).

(iv) $\mathcal{DB}$: Sends $C(R) = (g + 1, g^2)$ to $\mathcal{U}$.

(v) $\mathcal{U}$: Outputs $\mathsf{Dec}(sk, C(R)) - r = g^2 + g$ as $F(R)$, which is absurd (since $F(R) = g$).

Figure 3. An execution of the restricted version.

4.3 Failure probability

We have seen that the restricted version may not give $\mathcal{U}$ the expected result in Section 4.2. However, given the counterexample, we cannot conclude that the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement defined in Section 2.2. In fact, an EPIR protocol is said to be correct as long as it always gives $\mathcal{U}$ the expected result for any fixed input $(F(t), i) \in L[t] \times [n]$ *except with a negligible probability*. In other words, as a collection of probabilistic algorithms, an EPIR protocol is *allowed to fail with a negligible probability*. Therefore, to show that the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement, it is necessary to compute the failure probability of the protocol, i.e., the probability that the protocol does not give $\mathcal{U}$ the expected result.

In this section, we study the failure probability of the restricted version. We show, through experimental results, that the restricted version does fail with large probability for certain choices of $F(t)$ (e.g., $F(t) = g$).

From now on, we fix $p = 2$ to be the characteristic of all related finite fields. However, we stress that our methodology is applicable to any characteristic p. Following the notation of Sections 3.1 and 3.2, let $K = \mathrm{GF}(2)$ and $L = \mathrm{GF}(2^n)$ be the extension of K of degree n for an integer $n \geq 2$. Let $\mathbb{G} = L^\times$ be the multiplicative group of L of order $q = 2^n - 1$ and g be a generator of $\mathbb{G}$. Without loss of generality, we suppose $\alpha = g$. Then $G(t) = t \in K[t]$ is the polynomial of degree less than n such that $G(\alpha) = g$. For every $x \in \mathbb{Z}_q$, let $Y(t) \in K[t]$ be the polynomial of degree less than n such that $Y(\alpha) = y = g^x$. We define

$$D(t) = G(t)^x + Y(t) = t^x + Y(t) \in K[t].$$

Then the set of database blocks which satisfy the requirements imposed by Claim 4.1 (or in Section 3.2) is

$$\mathbb{D}_{n,g,x} = \{\beta \in \mathbb{G} : D(\beta) = 0\}.$$

We say that an execution of the restricted version is *parameterized* by (n, g, x, F, s, r, R) if $x \in \mathbb{Z}_q$ is the private key, $F(t) \in L[t]$ the polynomial to be evaluated, $s \in \mathbb{Z}_q$ and $r \in K$ the randomness used at step (ii) of the restricted version, and $R \in \mathbb{D}_{n,g,x}$ the database block held by $\mathcal{DB}$. Let $V(t), W(t) \in K[t]$ be the polynomials of degree less than n such that $V(g) = g^s$ and $W(g) = y^s(F(g)+r)$. Then the execution of the restricted version parameterized by (n, g, x, F, s, r, R) gives $\mathcal{U}$ the expected result if and only if $V(R) \neq 0$ and $E(R) = 0$, where

$$E(t) = W(t) + V(t)^x(F(t) + r). \tag{4.1}$$

For an execution of the restricted version parameterized by (n, g, x, F, s, r, R), we define

$$\mathbf{H}_{x,s,r,F,R} = \begin{cases} 1 & \text{if } V(R) \neq 0 \text{ and } E(R) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Then the execution fails if and only if $\mathbf{H}_{x,s,r,F,R} = 0$. Therefore, the probability that an execution of the restricted version fails when $x \in \mathbb{Z}_q$ is the private key and $F(t) \in L[t]$ is the polynomial chosen by $\mathcal{U}$ is exactly

$$\epsilon(n, g, x, F) = \Pr\big[s \leftarrow \mathbb{Z}_q, r \leftarrow K, R \leftarrow \mathbb{D}_{n,g,x} : \mathbf{H}_{x,s,r,F,R} = 0\big].$$

Since s, r and R are uniformly distributed, we have that

$$\epsilon(n, g, x, F) = \frac{\sum_{s\in\mathbb{Z}_q} \sum_{r\in K} \sum_{R\in\mathbb{D}_{n,g,x}} (1 - \mathbf{H}_{x,s,r,F,R})}{2q \cdot |\mathbb{D}_{n,g,x}|}. \tag{4.2}$$

n	$\mathrm{Min}_g(t)$	$\eta(n,g,g)$	n	$\mathrm{Min}_g(t)$	$\eta(n,g,g)$
2	t^2+t+1	0.61111	6	$t^6+t^4+t^3+t+1$	0.87719
3	t^3+t+1	0.74271	7	t^7+t+1	0.87895
4	t^4+t+1	0.81537	8	$t^8+t^4+t^3+t^2+1$	0.89809
5	t^5+t^2+1	0.83630	9	t^9+t^4+1	0.90358

Table 1. Failure probability.

The probability that the restricted version fails when $F(t) \in L[t]$ is the polynomial chosen by $\mathcal{U}$ is exactly

$$\eta(n,g,F) = \frac{1}{q} \sum_{x \in \mathbb{Z}_q} \epsilon(n,g,x,F). \tag{4.3}$$

The probabilities $\eta(n,g,F)$ for $2 \le n \le 9$ and $F(t) = g$ are quite large and enumerated in Table 1.

4.4 Bringer–Chabanne EPIR protocol fails frequently when $F(t) = g$

In this section, we show that the restricted version fails with large probability when $F(t) = g$. Specifically, for every integer $n \ge 2$, we give a lower bound on $\eta(n,g,g)$.

We follow the notation in Section 4.3. For every $j \in \mathbb{Z}_q$, we call the set $\mathbf{C} = \{j \cdot 2^k \bmod q : k = 0,1,2,\ldots\}$ a *cyclotomic coset* mod q. By default, $\mathbf{C}$ is represented by the smallest number $u \in \mathbf{C}$ and denoted as

$$\mathbf{C}_u = \{j \cdot 2^k \bmod q : k = 0,1,2,\ldots\}.$$

The number u is called the *coset representative* of $\mathbf{C}$. Clearly, all distinct cyclotomic cosets mod q are pairwise disjoint and form a partition of $\mathbb{Z}_q$, that is, $\mathbb{Z}_q = \bigcup_{u \in U} \mathbf{C}_u$, where U is the set of coset representatives of all distinct cyclotomic cosets mod q. For every positive integer d, we denote by $N_2(d)$ the number of monic irreducible polynomials of degree d in $K[t]$.

Lemma 4.2 (Lidl and Niederreiter [25]). *The following statements hold:*

(i) *For every $u \in U$, the cardinality of $\mathbf{C}_u$ is a divisor of n.*

(ii) *For every positive integer $d|n$, the number of cyclotomic cosets mod q of cardinality d is $N_2(d)$.*

(iii) *For every integer $d \ge 2$, we have that $N_2(d) \le \frac{1}{d}(2^d - 2)$.*

For every $u \in U$, we denote by

$$\mathbf{D}_u = \{g^j : j \in \mathbf{C}_u\}$$

the set of field elements in L which share the same *minimal polynomial* over K with g^u. For every $x \in \mathbb{Z}_q$, it is clear that there is a subset $U_x \subseteq U$ of coset representatives such that

$$\mathbb{D}_{n,g,x} = \bigcup_{u \in U_x} \mathbf{D}_u. \tag{4.4}$$

Lemma 4.3. *For every $x \in \mathbb{Z}_q$, we have that $1 \in U_x$.*

Proof. It follows from the fact that $D(t) \in K[t]$ and $D(g) = 0$. □

Due to (4.1), $E(t)$ is determined by the parameters $g \in \mathbb{G}$, $x \in \mathbb{Z}_q$, $F(t) \in L[t]$, $s \in \mathbb{Z}_q$ and $r \in K$. The next lemma shows that $E(t)$ and $D(t)$ only share a very small number of roots in L when $F(t) = g$.

Lemma 4.4. *Suppose $F(t) = g$. Then for every $x \in \mathbb{Z}_q$, $u \in U_x$, $s \in \mathbb{Z}_q$ and $r \in K$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$ or $E(t)$ has at most one root in $\mathbf{D}_u$.*

Proof. If $V(g^u) = 0$, then $V(g^{2^j \cdot u}) = V(g^u)^{2^j} = 0$ for any $j \in \mathbb{N}$, that is, $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$. Otherwise, we show that $E(t)$ has at most one root in $\mathbf{D}_u$. Due to (4.1), we have that

$$E(t) = W(t) + V(t)^x(g + r).$$

Suppose that $E(t)$ has two different roots in $\mathbf{D}_u$, say $g^{u \cdot 2^j}$ and $g^{u \cdot 2^k}$, where $0 \leq j < k < n$. Then

$$W(g^{u \cdot 2^j}) + V(g^{u \cdot 2^j})^x(g + r) = 0 = W(g^{u \cdot 2^k}) + V(g^{u \cdot 2^k})^x(g + r).$$

It follows that

$$(g + r)^{2^{n-j}} = (W(g^u)/V(g^u)^x)^{2^n} = (g + r)^{2^{n-k}}.$$

Since $r \in K$, the above equality implies $g^{2^{n-j}} = g^{2^{n-k}}$. Since g is primitive, we have $(2^n - 1) | (2^{n-j} - 2^{n-k})$. It follows that $n | (k - j)$, which is a contradiction. □

The following lemma gives a lower bound on $\epsilon(n, g, x, g)$ for any private key $x \in \mathbb{Z}_q$.

Lemma 4.5. *For every $x \in \mathbb{Z}_q$, we have that*

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|}.$$

Proof. Due to (4.2) and (4.4), we have that

$$\begin{aligned}\epsilon(n, g, x, g) &= \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{R\in\mathbb{D}_{n,g,x}}(1 - \mathbf{H}_{x,s,r,g,R})}{2q \cdot |\mathbb{D}_{n,g,x}|} \\ &= \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{u\in U_x}\sum_{R\in\mathbf{D}_u}(1 - \mathbf{H}_{x,s,r,g,R})}{2q \cdot |\mathbb{D}_{n,g,x}|}.\end{aligned}$$

Let $s \in \mathbb{Z}_q$ and $r \in K$ be arbitrary. Due to Lemma 4.4, for every $u \in U_x$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$, or $E(t)$ has at most one root in $\mathbf{D}_u$. It follows that

$$\sum_{R\in\mathbf{D}_u}(1 - \mathbf{H}_{x,s,r,g,R}) \geq |\mathbf{C}_u| - 1.$$

Therefore,

$$\epsilon(n, g, x, g) \geq \frac{\sum_{s\in\mathbb{Z}_q}\sum_{r\in K}\sum_{u\in U_x}(|\mathbf{C}_u| - 1)}{2q \cdot |\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|}. \qquad \square$$

We want to bound $\epsilon(n, g, x, g)$ for various settings of n and x. As the first case, we suppose that n is prime.

Lemma 4.6. *If n is prime, then $\epsilon(n, g, x, g) > 1 - \frac{2}{n}$ for every $x \in \mathbb{Z}_q$.*

Proof. Due to Lemma 4.2, $|\mathbf{C}_u|$ divides n for every $x \in \mathbb{Z}_q$ and $u \in U_x$. Since n is prime, we have that $|\mathbf{C}_u| = 1$ or n.

(i) If $|U_x| = 1$, then $U_x = \{1\}$ due to Lemma 4.3. It is obvious that $|\mathbf{C}_1| = n$. By Lemma 4.5, we have

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{1}{n} > 1 - \frac{2}{n}.$$

(ii) If $|U_x| > 1$ and $0 \in U_x$, then we have

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{1 + n(|U_x| - 1)} > 1 - \frac{2}{n}.$$

(iii) If $|U_x| > 1$ and $0 \notin U_x$, then we have

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{|U_x|}{n \cdot |U_x|} = 1 - \frac{1}{n} > 1 - \frac{2}{n}. \qquad \square$$

Below we lower bound $\epsilon(n, g, x, g)$ for *any* integer $n \geq 2$ and private key $x \in \mathbb{Z}_q$. For any positive integer $d \,|\, n$, we set

$$\lambda_{x,d} = \big|\{u : u \in U_x \text{ and } \mathbf{C}_u \text{ is of cardinality } d\}\big|.$$

Due to Lemma 4.3 and the requirements on database block R (imposed by Claim 4.1), $\lambda_x = (\lambda_{x,d})$ belongs to the set

$$\Psi_n = \big\{z = (z_d)_{d|n} : 0 \leq z_1 \leq 1;\ 1 \leq z_n \leq N_2(n);\\ 0 \leq z_d \leq N_2(d) \text{ for } d \,|\, n,\ 1 < d < n\big\},$$

where the coordinates of λ_x and z are indexed by positive divisors of n. Due to Lemma 4.5, we have that

$$\epsilon(n, g, x, g) \geq 1 - \frac{|U_x|}{|\mathbb{D}_{n,g,x}|} = 1 - \frac{\sum_{d|n} \lambda_{x,d}}{\sum_{d|n} d\lambda_{x,d}}. \tag{4.5}$$

We turn to upper bound the function

$$\psi_n(z) = \frac{\sum_{d|n} z_d}{\sum_{d|n} d z_d}$$

on Ψ_n. Because this is relatively hard, we upper bound the function

$$\phi_n(z) = \frac{\sum_{d=1}^{n} z_d}{\sum_{d=1}^{n} d z_d},$$

where $z = (z_1, \ldots, z_n)$ is taken from the set

$$\Phi_n = \big\{z = (z_1, \ldots, z_n) : 0 \leq z_1 \leq 1;\ 1 \leq z_n \leq N_2(n);\\ 0 \leq z_d \leq N_2(d) \text{ for } 1 < d < n\big\}.$$

Let $\omega(n)$ be the maximum value of $\phi_n(z)$ on Φ_n, i.e.,

$$\omega(n) = \max\{\phi_n(z) : z \in \Phi_n\}.$$

Lemma 4.7. *For every $x \in \mathbb{Z}_q$, we have that $\epsilon(n, g, x, g) \geq 1 - \omega(n)$.*

Proof. Clearly,

$$\omega(n) = \max\{\phi_n(z) : z \in \Phi_n\} \geq \max\{\psi_n(z) : z \in \Psi_n\} \geq \psi_n(\lambda_x).$$

Due to (4.5), we have that $\epsilon(n, g, x, g) \geq 1 - \psi_n(\lambda_x) \geq 1 - \omega(n)$ for every $x \in \mathbb{Z}_q$. □

Due to Lemma 4.7, it is sufficient to upper bound $\omega(n)$.

Lemma 4.8. *Suppose that $\omega(n) = \phi_n(\xi)$ for $\xi = (\xi_1, \ldots, \xi_n) \in \Phi_n$. Then $\xi_1 = \xi_n = 1$. Furthermore, if $n \geq 3$, then there is an integer $1 < h < n$ such that $\xi_d = N_2(d)$ for every integer $1 < d \leq h$ and $\xi_d = 0$ for every integer $h < d < n$.*

Proof. It is trivial to verify that $\xi_1 = \xi_2 = 1$ for $n = 2$. Let $n \geq 3$.

(i) For every $(0, z_2, \ldots, z_n), (1, z_2, \ldots, z_n) \in \Phi_n$, it is easy to see that

$$\phi_n(0, z_2, \ldots, z_n) - \phi_n(1, z_2, \ldots, z_n) < 0,$$

which implies that $\xi_1 = 1$.

(ii) For every $(1, z_2, \ldots, z_{n-1}, z_n), (1, z_2, \ldots, z_{n-1}, 1) \in \Phi_n$ (where $z_n > 1$), it is easy to see that

$$\phi_n(1, z_2, \ldots, z_{n-1}, z_n) - \phi_n(1, z_2, \ldots, z_{n-1}, 1) < 0,$$

which implies that $\xi_n = 1$.

(iii) Suppose $0 < \xi_h < N_2(h)$ for some integer $1 < h < n$. Let

$$C_1 = \sum_{d=1}^{h-1} \xi_d, \quad C_2 = \sum_{d=h+1}^{n} \xi_d, \quad C_3 = \sum_{d=1}^{h-1} d\xi_d, \quad C_4 = \sum_{d=h+1}^{n} d\xi_d.$$

Then due to the maximality of $\omega(n)$, we have

$$\begin{aligned} 0 &\geq \phi_n(\xi_1, \ldots, \xi_h + 1, \ldots, \xi_n) - \phi_n(\xi) \\ &= \frac{C_3 + C_4 - hC_1 - hC_2}{(C_3 + h(\xi_h + 1) + C_4)(C_3 + h\xi_h + C_4)} \end{aligned}$$

and

$$\begin{aligned} 0 &\geq \phi_n(\xi_1, \ldots, \xi_h - 1, \ldots, \xi_n) - \phi_n(\xi) \\ &= \frac{-C_3 - C_4 + hC_1 + hC_2}{(C_3 + h(\xi_h - 1) + C_4)(C_3 + h\xi_h + C_4)}. \end{aligned}$$

The above inequalities imply that $C_3 + C_4 = hC_1 + hC_2$. Hence, we have

$$h = \frac{\sum_{d=1}^{n} d\xi_d}{\sum_{d=1}^{n} \xi_d} = \frac{1}{\omega(n)}.$$

(iv) We claim that $\xi_a = N_2(a)$ for every $1 < a < h$. Otherwise, by (iii), we have that $\xi_a = 0$ and

$$\omega(n) < \phi_n(\xi_1, \ldots, \xi_a + 1, \ldots, \xi_h - 1, \ldots, \xi_n),$$

which is a contradiction.

(v) We claim that $\xi_b = 0$ for every $h < b < n$. Otherwise, by (iii), we have that $\xi_b = N_2(b)$ and

$$\omega(n) < \phi_n(\xi_1, \ldots, \xi_h + 1, \ldots, \xi_b - 1, \ldots, \xi_n),$$

which is a contradiction.

(vi) Finally, we show that $\omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1)$. Due to (iii), (iv) and (v), we have that

$$\xi = (1, N_2(2), \ldots, N_2(h-1), \xi_h, 0, \ldots, 0, 1),$$

where $0 < \xi_h < N_2(h)$. Since

$$\phi_n(\xi) = \omega(n) \geq \phi_n(1, N_2(2), \ldots, N_2(h-1), 0, 0, \ldots, 0, 1),$$

we have

$$hC_1 - C_3 \leq n - h.$$

If $hC_1 - C_3 < n - h$, then

$$\omega(n) < \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1),$$

which is a contradiction. Therefore, $hC_1 - C_3 = n - h$.Then it is not hard to verify that

$$\omega(n) = \phi_n(\xi) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1).$$

Therefore, we could have taken $\xi = (1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1)$. □

Due to Lemma 4.8, for every integer $n \geq 3$, there is at least one integer h with $1 < h < n$ such that

$$\omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1). \tag{4.6}$$

Note that the integer h may be not unique. For every integer $n \geq 3$, we define

$$h(n) = \min\{h : \omega(n) = \phi_n(1, N_2(2), \ldots, N_2(h), 0, \ldots, 0, 1), \text{ where } 1 < h < n\} \tag{4.7}$$

to be the smallest integer $1 < h < n$ such that (4.6) holds. The next lemma shows that $h(n)$ is an increasing function of n.

Lemma 4.9. *We have that $h(n+1) \geq h(n)$ for every integer $n \geq 3$.*

Proof. Due to the definition of $h(\cdot)$ by (4.7), it is not hard to see that

$$\phi_n(1, N_2(2), \ldots, N_2(l-1), N_2(l), 0, \ldots, 0, 1) > \phi_n(1, N_2(2), \ldots, N_2(l-1), 0, 0, \ldots, 0, 1)$$

for every integer $2 \leq l \leq h(n)$. Equivalently, we have that

$$\frac{1}{l} > \frac{\sum_{d=2}^{l-1} N_2(d) + 2}{\sum_{d=2}^{l-1} dN_2(d) + n + 1} \tag{4.8}$$

for every integer $2 \leq l \leq h(n)$. Due to (4.8), it is not hard to verify that

$$\phi_{n+1}(1, N_2(2), \ldots, N_2(l-1), N_2(l), 0, \ldots, 0, 1) > \phi_{n+1}(1, N_2(2), \ldots, N_2(l-1), 0, 0, \ldots, 0, 1) \tag{4.9}$$

for every integer $2 \leq l \leq h(n)$. In particular, (4.9) holds for $l = h(n)$. This implies that $h(n+1) \geq h(n)$. □

On the other hand, $\omega(n)$ is a decreasing function of n:

Lemma 4.10. *We have that $\omega(n+1) < \omega(n)$ for every integer $n \geq 3$.*

Proof. By Lemma 4.9, we have that $h(n+1) \geq h(n)$. If $h(n+1) = h(n)$, then

$$\begin{aligned}\omega(n+1) &= \frac{\sum_{d=2}^{h(n+1)} N_2(d) + 2}{\sum_{d=2}^{h(n+1)} dN_2(d) + n + 2} \\ &= \frac{\sum_{d=2}^{h(n)} N_2(d) + 2}{\sum_{d=2}^{h(n)} dN_2(d) + n + 2} \\ &< \frac{\sum_{d=2}^{h(n)} N_2(d) + 2}{\sum_{d=2}^{h(n)} dN_2(d) + n + 1} = \omega(n).\end{aligned}$$

If $h(n+1) > h(n)$, then

$$\begin{aligned}\omega(n) &= \frac{\sum_{d=2}^{h(n)} N_2(d) + 2}{\sum_{d=2}^{h(n)} dN_2(d) + n + 1} \\ &\geq \frac{1}{h(n)+1} \geq \frac{1}{h(n+1)} \\ &> \frac{\sum_{d=2}^{h(n+1)} N_2(d) + 2}{\sum_{d=2}^{h(n+1)} dN_2(d) + n + 2} = \omega(n+1),\end{aligned}$$

where the first and the third inequality follow from the definition of $h(\cdot)$ by (4.7). □

We enumerate the values of $h(n)$ and $\omega(n)$ for some integers n in Table 2.

Lemma 4.11. *For every integer $n \geq 7$, we have that $\omega(n) \geq \frac{5}{n+9}$.*

Proof. Due to Table 2 and Lemma 4.9, we have that $h(n) \geq 3$ for every integer $n \geq 7$. It follows that $\omega(n) \geq \phi_n(1, 1, 2, 0, \ldots, 0, 1) = 5/(n+9)$. □

At last, we have the following theorem.

Theorem 4.12. *We have that*

$$\eta(n, g, g) \geq \begin{cases} 1 - \omega(n) & \text{if } 2 \leq n \leq 6 \text{ or } n \geq 7 \text{ is composite,} \\ 1 - \frac{2}{n} & \text{if } n \geq 7 \text{ is prime.} \end{cases}$$

n	$h(n)$	$\omega(n)$	n	$h(n)$	$\omega(n)$	n	$h(n)$	$\omega(n)$
2	1	0.66667	12	4	0.24242	296	10	0.09996
3	1	0.50000	20	5	0.19718	522	11	0.09089
4	2	0.42857	34	6	0.16547	934	12	0.08332
5	2	0.37500	57	7	0.14236	1681	13	0.07692
6	2	0.33333	98	8	0.12478	3058	14	0.07143
7	3	0.31250	169	9	0.11101	5596	15	0.06667

Table 2. The values of $h(n)$ and $\omega(n)$.

Proof. Table 2 shows that $\omega(n) \leq 2/n$ for every integer $2 \leq n \leq 6$. Due to Lemma 4.6 and Lemma 4.7, we have that

$$\epsilon(n, g, x, g) \geq \max\{1 - 2/n, 1 - \omega(n)\} = 1 - \omega(n)$$

for $n = 2, 3, 5$, and $\epsilon(n, g, x, g) \geq 1 - \omega(n)$ for $n = 4, 6$. Due to (4.3), we have that

$$\eta(n, g, g) = \frac{1}{q} \sum_{x \in \mathbb{Z}_q} \epsilon(n, g, x, g) \geq 1 - \omega(n).$$

Due to Lemmas 4.6, 4.7 and 4.11, we have that

$$\epsilon(n, g, x, g) \geq \max\{1 - 2/n, 1 - \omega(n)\} = 1 - 2/n$$

if $n \geq 7$ is prime and $\epsilon(n, g, x, g) \geq 1 - \omega(n)$ if $n \geq 7$ is composite. Due to (4.3), we have that $\eta(n, g, g) \geq 1 - 2/n$ if $n \geq 7$ is prime and $\eta(n, g, g) \geq 1 - \omega(n)$ if $n \geq 7$ is composite. □

By Theorem 4.12, Lemma 4.10 and Table 2, we see that $\eta(n, g, g)$ is always non-negligible. Hence, we have the following theorem.

Theorem 4.13. *The restricted version does not satisfy the correctness requirement if $F(t) = g$.*

4.5 Extension to a set of polynomials

In this section, we extend Theorem 4.13 to a set of polynomials $F(t) \in L[t]$. In particular, we follow the notation in Section 4.4 and show that the restricted

version does not satisfy the correctness requirement if $F(t) \in \mathcal{P}$, where

$$\mathcal{P} = \Big\{ f(t) = \sum_{k=0}^{d} f_k t^k : \exists\, 0 \le l \le d \text{ such that } f_l \in L \text{ is primitive and } f_k \in K \text{ for every } k \ne l \Big\}.$$

Note that the polynomial $F(t) = g \in L[t]$ we studied in Section 4.4 is in $\mathcal{P}$ and satisfies Lemma 4.4, which is critical for obtaining all subsequent lemmas and theorems. The next lemma shows that Lemma 4.4 holds for any polynomial $F(t) \in \mathcal{P}$ as well.

Lemma 4.14. *Let $F(t) \in \mathcal{P}$. Then for every $x \in \mathbb{Z}_q$, $u \in U_x$, $s \in \mathbb{Z}_q$ and $r \in K$, either $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$ or $E(t)$ has at most one root in $\mathbf{D}_u$.*

Proof. If $V(g^u) = 0$, then $V(g^{u \cdot 2^j}) = V(g^u)^{2^j} = 0$ for every $j \in \mathbb{N}$, that is, $V(\beta) = 0$ for every $\beta \in \mathbf{D}_u$. Otherwise, we have $V(\beta) \ne 0$ for every $\beta \in \mathbf{D}_u$. Suppose $F(t) = \sum_{k=0}^{d} F_k t^k$, where $F_l \in L$ is of order q and $F_k \in K$ for every $k \ne l$. We show that $E(t)$ has at most one root in $\mathbf{D}_u$, where

$$E(t) = W(t) + V(t)^x (F(t) + r).$$

Suppose $E(t)$ has two different roots in $\mathbf{D}_u$, say $g^{u \cdot 2^a}$ and $g^{u \cdot 2^b}$, where $0 \le a < b < n$. Then

$$W(g^{u \cdot 2^a}) + V(g^{u \cdot 2^a})^x (F(g^{u \cdot 2^a}) + r) = 0 = W(g^{u \cdot 2^b}) + V(g^{u \cdot 2^b})^x (F(g^{u \cdot 2^b}) + r).$$

It follows that

$$(F(g^{u \cdot 2^a}) + r)^{2^{n-a}} = (F(g^{u \cdot 2^b}) + r)^{2^{n-b}}. \tag{4.10}$$

Let $c \in \{a, b\}$. Then it is not hard to see that

$$(F(g^{u \cdot 2^c}) + r)^{2^{n-c}} = \sum_{k=0}^{l-1} F_k g^{uk} + \sum_{k=l+1}^{d} F_k g^{uk} + F_l^{2^{n-c}} g^{ul} + r.$$

Due to (4.10), we have that $F_l^{2^{n-a}} = F_l^{2^{n-b}}$. Since $F_l \in L$ is primitive, we have $(2^n - 1) | (2^{n-a} - 2^{n-b})$ and therefore $n | (b - a)$, which is a contradiction. □

Due to Lemma 4.14, we note that all lemmas and theorems subsequent to Lemma 4.4 in Section 4.4 can be generalized for any polynomial $F(t) \in \mathcal{P}$. Therefore, we deduce the following result:

Theorem 4.15. *The restricted version does not satisfy the correctness requirement if* $F(t) \in \mathcal{P}$.

4.6 Extension to any characteristic $p > 2$

We have stressed in Section 4.3 that our methodology is applicable when the characteristic of all related finite fields is any prime p. For example, it is obvious that we have an analog of Lemma 4.7 for any characteristic $p > 2$. Let $\omega_p(n)$ be an analog of the function $\omega(n)$ when the characteristic of all related finite fields is a prime $p > 2$. Then the following theorem holds as well.

Theorem 4.16. *We have that* $\eta(n, g, g) \geq 1 - \omega_p(n)$ *for every integer* $n \geq 2$, *where* $g \in \mathrm{GF}(p^n)$ *is primitive and* p *is an arbitrary prime number.*

It follows that Theorem 4.15 also holds when the characteristic of all related finite fields is any prime $p > 2$.

5 Conclusion

In this paper, we show that the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement. To simplify the argument, we give a restricted version of the Bringer–Chabanne EPIR protocol. If the original protocol satisfies the correctness requirement, then so does the restricted version. We show that the restricted version fails frequently if the polynomial to be evaluated has some special property. This allows us to get the expected conclusion, i.e., the Bringer–Chabanne EPIR protocol does not satisfy the correctness requirement.

Bibliography

[1] A. Ambainis, Upper bound on the communication complexity of private information retrieval, in: *ICALP 1997*, Lecture Notes in Comput. Sci. 1256, Springer, Heidelberg (1997), 401–407.

[2] A. Beimel, Y. Ishai and E. Kushilevitz, General constructions for information-theoretic private information retrieval, *Journal of Computer and System Sciences* **71** (2005), 213–247.

[3] A. Beimel, Y. Ishai, E. Kushilevitz and J. F. Raymond, Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval, in: *FOCS 2002*, IEEE, Los Alamitos (2002), 261–270.

[4] G. Brassard, C. Crépeau and J. M. Robert, All-or-nothing disclosure of secrets, in: *CRYPTO 1986*, Lecture Notes in Comput. Sci. 263, Springer, Heidelberg (1987), 234–238.

[5] J. Bringer and H. Chabanne, Another look at extended private information retrieval protocols, in: *AFRICACRYPT 2009*, Lecture Notes in Comput. Sci. 5580, Springer, Heidelberg (2009), 305–322.

[6] J. Bringer, H. Chabanne, D. Pointcheval and Q. Tang, Extended private information retrieval and its application in biometrics authentications, in: *CANS 2007*, Lecture Notes in Comput. Sci. 6467, Springer, Heidelberg (2007), 175–193.

[7] C. Cachin, S. Micali and M. Stadler, Computationally private information retrieval with polylogarithmic communication, in: *EUROCRYPT 1999*, Lecture Notes in Comput. Sci. 1592, Springer, Heidelberg (1999), 402–414.

[8] R. Canetti, Y. Ishai, R. Kumar, M. K. Reiter, R. Rubinfeld and R. N. Wright, Selective private function evaluation with applications to private statistics, In: *PODC 2001*, Association for Computing Machinery, New York (2001), 293–304.

[9] Y. C. Chang, Single database private information retrieval with logarithmic communication, in: *ACISP 2004*, Lecture Notes in Comput. Sci. 3108, Springer, Heidelberg (2004), 50–61.

[10] Y. M. Chee, T. Feng, S. Ling, H. Wang and L. F. Zhang, Query-efficient locally decodable codes of subexponential length, preprint (2010), `http://arxiv.org/abs/1008.1617`.

[11] B. Chor and N. Gilboa, Computationally private information retrieval, in: *STOC 1997*, Association for Computing Machinery, New York (1997), 304–313.

[12] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, Private information retrieval, in: *FOCS 1995*, IEEE, Los Alamitos (1995), 41–50.

[13] K. Efremenko, 3-query locally decodable codes of subexponential length, in: *STOC 2009*, Association for Computing Machinery, New York (2009), 39–44.

[14] S. Even, O. Goldreich and A. Lempel, A randomized protocol for signing contracts, *Communications of the Association for Computing Machinery* **28** (1985), 637–647.

[15] M. J. Freedman, Y. Ishai, B. Pinkas and O. Reingold, Keyword search and oblivious pseudorandom functions, in: *TCC 2005*, Lecture Notes in Comput. Sci. 3378, Springer, Heidelberg (2005), 303–324.

[16] C. Gentry and Z. Ramzan, Single-database private information retrieval with constant communication rate, in: *ICALP 2005*, Lecture Notes in Comput. Sci. 3580, Springer, Heidelberg (2005), 803–815.

[17] Y. Gertner, Y. Ishai, R. Kushilevitz and T. Malkin, Protecting data privacy in private information retrieval schemes, in: *STOC 1998*, Association for Computing Machinery, New York (1998), 151–160.

[18] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2, Cambridge University Press, Cambridge, 2004.

[19] O. Goldreich, S. Micali and A. Wigderson, How to play any mental game or a completeness theorem for protocols with honest majority, in: *STOC 1987*, Association for Computing Machinery, New York (1987), 218–229.

[20] J. Groth, A. Kiayias and H. Lipmaa, Multi-query computationally-private information retrieval with constant communication rate, in: *PKC 2010*, Lecture Notes in Comput. Sci. 6056, Springer, Heidelberg (2010), 107–123.

[21] T. Itoh and Y. Suzuki, New constructions for query-efficient locally decodable codes of subexponential length, *IEICE Transactions on Information and Systems* **E93-D** (2010), 263–270.

[22] J. Kilian, Founding cryptography on oblivious transfer, in: *STOC 1988*, Association for Computing Machinery, New York (1988), 20–31.

[23] E. Kushilevitz and R. Ostrovsky, Replication is not needed: single database, computationally-private information retrieval, in: *FOCS 1997*, IEEE, Los Alamitos (1997), 364–373.

[24] E. Kushilevitz and R. Ostrovsky, One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval, in: *EUROCRYPT 2000*, Lecture Notes in Comput. Sci. 1807, Springer, Heidelberg (2000), 104–121.

[25] R. Lidl and H. Niederreiter, *Finite Fields*, 2nd ed., Cambridge University Press, 1997.

[26] H. Lipmaa, An oblivious transfer protocol with log-squared communication, in: *ISC 2005*, Lecture Notes in Comput. Sci. 3650, Springer, Heidelberg (2005), 314–328.

[27] M. Naor and B. Pinkas, Oblivious transfer and polynomial evaluation. in: *STOC 1999*, Association for Computing Machinery, New York (1999), 245–254.

[28] M. O. Rabin, *How to exchange secrets by oblivious transfer*, Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[29] D. P. Woodruff and S. Yekhanin, A geometric approach to information-theoretic private information retrieval, in: *CCC 2005*, IEEE, Los Alamitos (2005), 275–284.

[30] A. Yamamura and T. Saito, Private information retrieval based on the subgroup membership problem, in: *ACISP 2001*, Lecture Notes in Comput. Sci. 2119, Springer, Heidelberg (2001), 206–220.

[31] S. Yekhanin, Towards 3-query locally decodable codes of subexponential length, in: *STOC 2007*, Association for Computing Machinery, New York (2007), 266–274.

Received May 4, 2011; revised January 16, 2012.

Author information

Yeow Meng Chee, Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, 637371, Singapore.
E-mail: ymchee@ntu.edu.sg

Huaxiong Wang, Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, 637371, Singapore.
E-mail: hxwang@ntu.edu.sg

Liang Feng Zhang, Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, 637371, Singapore.
E-mail: liangfeng.zhang@ntu.edu.sg