Efficient and Secure Equality Tests

Majid Nateghizad, Zekeriya Erkin and Reginald L. Lagendijk
Cyber Security Group, Department of Intelligent Systems
Delft University of Technology, The Netherlands
Email: {m.nateghizad, z.erkin, r.l.lagendijk}@tudelft.nl

***Abstract*—Secure equality testing of two private values is one of the fundamental building blocks of many cryptographic protocols designed for Signal Processing in the Encrypted Domain (SPED). Existing protocols introduce significant amount of computation and computational overhead, which makes it essential to search for new and novel, efficient equality tests for the design of SPED algorithms. In this paper, we first describe the state-of-the-art equality tests, and then propose two cryptographic protocols which are significantly more efficient than the existing work. Our proposals achieve high performance due to algorithmic changes and successful deployment of data packing. Furthermore, we also present a novel secure exponentiation protocol as a part of our first equality test. Complexity and performance analyses clearly indicate the high efficiency of our protocols in terms of computation cost.**

I. Introduction

RECENT advances in the collection, processing, and delivery of digital contents have been deployed in many domains. However, processing sensitive information have also raised several privacy concerns. Possible misuse or leakage of privacy-sensitive data has several consequences. Therefore, privacy research has been one of the most attractive topics in the last few years. SPED, as one of the solutions to preserve the privacy of users, has found many applications in several fields. Biometric data matching [1], [2], recommender systems [3], [4], data mining [5], [6], and data aggregation [7], [8] are only a few examples. Instead of implicitly assuming that all processing parties are trusted, SPED provides an environment for the parties, e.g. server and client, to collaborate for processing privacy-sensitive information in a privacy-preserving manner. The main idea in SPED is to provide only the encrypted version of the data to the server, and invoke interactive cryptographic protocols between the server and the decryption key owner to perform the desired algorithms.

Although SPED protects the privacy of sensitive information, its computational and communication overhead are the main challenges in large scale applications. Actually, the SPED algorithms are attractive from privacy preservation perspective but they are far more complex than their plaintext versions since the core operations like comparison, equality tests and division, which are repeated in large quantities, are computationally demanding. Therefore, reducing the complexity of such operations is an important challenge to make the cryptographic solutions practical. As a result, a number of protocols have been introduced [9], [10], [11], [12].

In this work, we address secure equality testing (EQT), one of the fundamental operations needed in many SPED solutions, e.g. for search algorithms over encrypted data [13], which is addressed previously in [14], [11], [15]. In our scenario, Alice holds two encrypted values while Bob has the decryption key. Our aim is to design a cryptographic protocol for Alice and Bob that outputs a single encrypted bit, the result of the equality test, which is also secret for both. We propose two EQT protocols based on [11], [15]. We introduce algorithmic changes and data packing that improve the performance of the existing work significantly as shown in the complexity analysis. Experimental results also show that our protocols outperform the existing work, and present up to 99% run-time improvement in a fair experimental setup.

The rest of the paper is organized as follows. We introduce the notation, the cryptographic building blocks, and the security assumptions in Section II. In Section III, we describe the related work. We describe our proposals, two EQT protocols, in Section IV, and provide complexity and security analyses in Section V. We demonstrate the performance of the proposals in terms of computational complexity and run-time in Section VI. Finally, we conclude in Section VII.

II. Preliminaries

In this section, we describe the application setting, the security assumptions, and the cryptographic tools used in this article. We summarize our notation in Table I.

A. Homomorphic Encryption

In this article, we rely on an additively homomorphic cryptosystem, more specifically the Paillier cryptosystem [16]. An additively homomorphic encryption scheme preserves certain structure that can be exploited to process ciphertexts without decryption. Given $\mathcal{E}_{pk}(m_1)$ and $\mathcal{E}_{pk}(m_2)$, a new ciphertext whose decryption yields the sum of the plaintext messages m_1 and m_2 can be obtained by performing a multiplication operation over the ciphertexts under additively homomorphic encryption schemes: $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2)) = m_1 + m_2$.

Consequently, exponentiation of any ciphertext with a public value yields the encrypted product of the original plaintext and the exponent: $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)^e) = e \cdot m$.

In the rest of the paper, we denote the ciphertext of a message m by $[m]$ for the Paillier cryptosystem.

B. Security Assumptions and the Setting

We consider the semi-honest security model, where parties are honest in following the protocol steps, while they can keep messages to deduce more information than they are entitled

978-1-5090-1138-4/16/$31.00 ©2016 IEEE

TABLE I
SYMBOLS AND THEIR MEANING.

Symbol	Description	Symbol	Description	Symbol	Description
a, b	input messages	ρ	number of plaintext can	$d, \grave{d}$	Hamming distance
x_i	the i^{th} bit of integer x		packed into one ciphertext	$[.]$	Paillier encryption
ℓ	bit length of inputs	n	crypto system modulus	α_i	coefficients of Lagrange polynomial
κ	statistical security parameter	$\binom{i}{e}$	binomial coefficients	$\oplus$	exclusive-OR
$r, \grave{r}, R$	random numbers	ϑ	result of the equality test	N	number of equality tests to be performed
pk	public key	sk	private key	$u_{(\lambda)}$	u exponentiations with λ-bit exponents

to. We assume that Bob has the decryption key sk and Alice has the encryptions of two values a and b, which are ℓ-bit integers each. The values a and b are secret and should be kept hidden from Alice and Bob. However, Alice should obtain an encrypted bit $\vartheta \leftarrow (a = b) ? 1 : 0$, where ϑ is the result of the equality test which is also kept secret from Alice and Bob. During the computation of ϑ, the intermediate values should also be kept secret from both parties to limit the information leakage.

III. RELATED WORK

The ideas behind existing equality testing include using Hamming distance, quadratic residuosity assumption, and bit-decomposition. Takashi and Kazuo [14] proposed a probabilistic constant-round equality test protocol, where Jacobi symbol is used to test quadratic residuosity of a value. Schoenmakers and Tuyls [11] have shown a practical method to check the equality of two encrypted values by introducing a bit-decomposition protocol. In [15], Lipmaa and Toft have introduced an equality test protocol, which is based on computing Hamming distance between two private values. The protocol uses Lagrange interpolation [17] to generate a polynomial, which is used to obtain the result of the equality test, and a multiplicative masking method described in [18]. In the following, we present [15] and [11] that inspired us to design our protocols, which are also deterministic unlike [14].

A. EQT based on the Hamming distance (LT13)

The equality testing protocol in [15] is based on computing the Hamming distance of two private values. LT13 computes a polynomial in order to obtain an encrypted bit, which represents the result of the equality testing protocol as given in Protocol 1.

B. EQT based on the bit-decomposition (ST06)

The protocol in [11] decomposes $[z] \leftarrow [a - b]$ to its encrypted bits $[z_i]$, $0 \leq i < \ell$, and then computes $[\prod_{i=0}^{\ell-1}(1 - z_i)]$, which is $[1]$ in case of equality, by using a secure multiplication protocol as summarized in Protocol 2.

IV. IMPROVED SECURE EQUALITY TESTS

We now describe two equality testing protocols based on LT13 and ST06. We achieve this by introducing a novel secure exponentiation protocol and employing data packing.

Protocol 1 LT13

1: Alice computes $[a - b]$, additively masks the result with a random number r, and sends $[x] \leftarrow [a - b + r]$ to Bob.
2: Bob decrypts the message $[x]$ and computes the first ℓ bits x_i, encrypts them separately, and sends them to Alice.
3: Alice computes the Hamming distance $[d]$ between $[x_i]$ and $[r_i]$, masks it, and returns the masked encrypted value $[y]$ to Bob. The Hamming distance is zero if and only if $a = b$, since in this case $x = r$. Note that d is masked multiplicatively with the inverse of a random number r.
4: Bob decrypts $[y]$ and computes the exponentiations y^i, $1 < i \leq \ell$. The y^i are encrypted and sent back to Alice.
5: Alice unmasks the $[y^i]$ to obtain $[d^i]$ and computes ℓ-degree Lagrange polynomial $[\vartheta] \leftarrow [\sum_{i=0}^{\ell} \alpha_i \cdot d^i]$, where α_i are coefficients that depend on ℓ.

Protocol 2 ST06

1: Alice computes $[z] \leftarrow [a - b]$, masks the result additively, and sends $[x] \leftarrow [z + r]$ to Bob.
2: Bob decrypts $[x]$, computes the first ℓ bits, encrypts them separately, and sends $[x_i]$, $0 \leq i < \ell$, to Alice.
3: Alice runs a secure subroutine, based on a secure multiplication protocol, to compute the first ℓ bits of $[z]$ (as explained in Protocol 6), namely $[z_i]$, $0 \leq i < \ell$.
4: Alice runs another subroutine to compute $[\vartheta] \leftarrow [\prod_{i=0}^{\ell-1}(1 - z_i)]$.

A. Improved EQT based on the Hamming distance (NEL-I)

To improve LT13 in terms of computational complexity, first, we introduce a novel method of computing exponentiation securely, then we use data packing [19] to decrease the decryption cost [10]. Note that packing technique can be used when we have multiple equality tests to perform at once. Furthermore, we add one more round to the protocol, which results in a significant decrease in computational complexity. In fact, the degree of the polynomial that Alice has to compute decreases from ℓ to $\lceil \log_2 \ell \rceil$.

In LT13, Alice computes the Hamming distance d between x and r, and then generates the polynomial based on d, which results in generating an ℓ-degree polynomial. Computing an ℓ-degree polynomial over encrypted data introduces significant computational overhead. To decrease the degree of the polynomial, we add one more round in NEL-I : As shown in Protocol 3, Alice computes the Hamming distance $[d]$, then she masks it with a random value $[y] \leftarrow [d + \grave{r}]$. Afterwards, she computes the Hamming distance $[\grave{d}]$ between $[y]$ and $[\grave{r}]$. Since $0 \leq \grave{d} \leq \lceil \log_2 \ell \rceil$, Alice constructs a $\lceil \log_2 \ell \rceil$-degree polynomial using the specified mapping and computes the values $[\grave{d}^i]$, which are used later to obtain $[\vartheta]$.

Protocol 3 NEL-I

```
1: Alice generates a sufficiently large ($\ell + \kappa$ bits) random value $r$, computes $[x] \leftarrow [a - b + r]$ and sends $[x]$ to Bob. Alice puts multiple $x$ values into a single ciphertext by using data packing. Assume that $(a, b)$ and $r$ are $\ell$ and $\ell+\kappa$ bits integers, respectively. Then, $[x]$ is a $(\ell+\kappa+1)$-bit integer. Let the message space of the Paillier cryptosystem be $n$, then Alice packs $\rho = \lfloor n/(\ell+\kappa+1) \rfloor$ into one Paillier message as follows:
   $$[\hat{x}] = \sum_{j=0}^{\rho-1} [x_j \cdot (2^{\ell+\kappa+1})^j] \,, \tag{1}$$
   and then sends $[\hat{x}]$ to Bob.
2: Bob decrypts $[\hat{x}]$ and unpacks it. Then, he computes the first $\ell$ bits $x_i$, for $0 \leq i < \ell$, of each component; encrypts them separately and sends $[x_i]$ to Alice.
3: Alice computes $[r_i \oplus x_i]$ for $0 \leq i < \ell$ and then $[d] \leftarrow [\sum_{i=0}^{\ell-1} r_i \oplus x_i] = \prod_{i=0}^{\ell-1} [r_i \oplus x_i]$, which is the Hamming distance of $r$ and $x$ (note that $[r_i \oplus x_i] \leftarrow [x][r][x]^{-2r}$). Then, she additively masks $[d]$ with an $(\lceil \log_2 \ell \rceil + \kappa)$-bit random number $\grave{r}$ that is $[y] \leftarrow [d + \grave{r}]$ and sends $[y]$ to Bob in packed form $[\hat{y}]$.
4: Bob decrypts $[\hat{y}]$ and unpacks it. Then, he computes the first $\lceil \log_2 \ell \rceil$ bits $y_i$, $0 \leq i < \lceil \log_2 \ell \rceil$ of each component, encrypts them separately, and sends the $[y_i]$ to Alice.
5: Alice computes $[\grave{r}_i \oplus y_i]$ for $0 \leq i < \lceil \log_2 \ell \rceil$ and then $[\grave{d}] \leftarrow [\sum_{i=0}^{\lceil \log_2 \ell \rceil - 1} \grave{r}_i \oplus y_i] = \prod_{i=0}^{\lceil \log_2 \ell \rceil - 1} [\grave{r}_i \oplus y_i]$. Then, she additively masks $[\grave{d}]$ with another $(\lceil \log_2 \log_2 \ell \rceil + \kappa)$-bit random number $R$, computes $[t] \leftarrow [\grave{d} + R]$, and sends $[t]$ to Bob in packed form $[\hat{t}]$.
6: Bob decrypts $[t]$, computes $t^i$, $1 < i \leq \lceil \log_2 \ell \rceil$, and sends $t^i$ to Alice in encrypted form.
7: Alice unmasks $[t^i]$ by computing $[\grave{d}^i] \leftarrow [t^i - \sum_{e=1}^{i} \binom{i}{e} \grave{d}^{i-e} R^e] = [t^i][\prod_{e=1}^{i} [\grave{d}^{i-e}]^{\binom{i}{e} R^e}]$, $1 < i \leq \lceil \log_2 \ell \rceil$.
8: Alice constructs a $\lceil \log_2 \ell \rceil$-degree polynomial $[\vartheta] \leftarrow [\sum_{i=0}^{\lceil \log_2 \ell \rceil} \alpha_i \cdot \grave{d}^i] = \prod_{i=0}^{\lceil \log_2 \ell \rceil} [\grave{d}^i]^{\alpha_i}$, where it maps $\grave{d} = 1$ to 1, and $\grave{d} \in \{2, 3, \ldots, \lceil \log_2 \ell \rceil\}$ to 0.
```

Another computationally demanding part of LT13 is the secure exponentiation method, which is required in order to compute $[\vartheta] \leftarrow [\sum_{i=0}^{c} \alpha_i \cdot d^i]$. In the secure exponentiation protocol, Alice has $[d]$ and she has to compute $[d^i]$ with Bob's help, who has the private key. Although the secure exponentiation method in LT13, given in Protocol 4, is simple and straightforward, unmasking, $[d^i] \leftarrow [t^i]^{R^i}$, is very expensive. In fact, the unmasking dominates the overall computational complexity of LT13. The unmasking in Protocol 3 is an extended form of a method used in [2], which is also computationally expensive. In this work, we introduce a novel secure exponentiation method, Protocol 5, which makes the unmasking significantly less expensive. As it is shown in Protocol 5 (which replaces Step 7 in Protocol 3), the additive masking is used instead of multiplicative form to blind $[d]$. Then, we use an efficient method to unmask the encrypted values and obtain $[d^i]$. Note that unlike the secure exponentiation in LT13, where $[d^\ell]$ can be obtained directly, we need to compute all $[d^i]$, $1 < i < \ell$, before computing $[d^\ell]$ in NEL-I. However, both LT13 and NEL-I demand computation of $[\sum_{i=0}^{\ell} \alpha_i \cdot d^i]$, which requires computing all $[d^i]$, $1 < i \leq \ell$.

Protocol 4 Secure exponentiation (LT13)

```
Input: $[d]$
Output: $[d^i]$ for $1 < i \leq \ell$
1: Alice chooses a random number $R \in \mathbb{Z}_N^*$, computes its inverse $R^{-1}$, and $R^i$, $1 < i \leq \ell$. Then, Alice multiplicatively masks $[d] \leftarrow [d + 1]$ with $R^{-1}$ that is $[t] \leftarrow [d + 1]^{R^{-1}}$ and sends $[t]$ to party B. Note that 1 is added to $[d]$ to make sure $d \in \mathbb{Z}_{N^2}^*$, which is needed to get the correct result after performing unmasking.
2: Party B decrypts $[t]$, computes $t^i$, $1 < i \leq \ell$, and sends $t^i$ to party A in encrypted form.
3: Party A unmasks $[t^i]$ by computing $[d^i] \leftarrow [t^i]^{R^i}$ for $1 < i \leq \ell$.
```

Protocol 5 Secure exponentiation (NEL-I)

```
Input: $[d]$
Output: $[d^i]$ for $1 < i \leq \ell$
1: Alice chooses a random number $R$, where $R$ is a $(\lceil \log_2 d \rceil + \kappa)$-bit value, and then she sends $[t] \leftarrow [d + R]$ to Bob.
2: Party B decrypts $[t]$, computes $t^i$, $1 < i \leq \ell$, and sends $t^i$ to party A in encrypted form.
3: Alice has $[d]$ and $R$ values and she can easily compute $[p_1] \leftarrow [dR]$, and then $[d^2] \leftarrow [t^2 - 2p_1 - R^2]$. To compute $[d^3]$, she computes $[p_1] \leftarrow [d^2 R]$ and $[p_2] \leftarrow [p_1 R]$, and then $[d^3] \leftarrow [t^3 - \sum_{e=1}^{2} \binom{3}{e} p_e - R^3]$. In order to obtain $[d^\ell]$, she computes $[p_1] \leftarrow [d^{\ell-1} R]$ and $[p_i] \leftarrow [p_{i-1} R]$, $1 < i < \ell$, and then she computes $[d^\ell] \leftarrow [t^\ell - \sum_{e=1}^{\ell-1} \binom{\ell}{e} p_e - R^\ell]$.
```

B. *Improved EQT based on the bit-decomposition (NEL-II)*

Ignoring the decryption cost, ST06 has a very low computational complexity compared to LT13. However, the number of times that Bob invokes decryption in ST06 is very high, which makes the protocol computationally expensive. To improve the performance, we propose a variant of ST06 that is also based on the bit-decomposition method but employs data packing, introducing a significant improvement in computation. We provide the details in Protocol 6.

V. Security analysis

In this section, we argue that our algorithmic changes and deploying packing technique do not violate the security of

Protocol 6 NEL-II

```
1: Alice computes $[z] \leftarrow [a-b]$, $[x] \leftarrow [z+r]$, where $r$ is a $(\ell+1+\kappa)$-bit random number, and sends $[x]$ to Bob in the packed form.
2: Bob decrypts and unpacks $[x]$, decomposes first $\ell$ bits $x_i$, $0 \leq i < \ell$, and sends $[x_i]$ to Alice.
3: Alice computes $[c_0] \leftarrow [x_0]^{r_0}$, $[z_0] \leftarrow [x_0][r_0][c_0]^{-2}$, and sets $i = 1$.
4: Alice chooses $\ell - 2$ random bits $R$ to mask $[c_{i-1}]$ such that $[\theta] \leftarrow [c_{i-1} \oplus R]$. Afterward, Alice sends $[\theta]$ to Bob. Note that the $[\theta]$ are encrypted one-bit values, which means Alice can pack $n$ messages into one ciphertext, $\rho = n$ that decreases the Paillier decryption and communication costs significantly.
       $$[\hat{\theta}] = \sum_{j=0}^{n-1} [\theta_j \cdot 2^j] \, , \qquad (2)$$
5: Bob decrypts and unpacks the $[\hat{\theta}]$ to obtain $\theta$, then computes $\alpha \leftarrow \theta \times x_i$ and sends $[\alpha]$ to Alice.
6: Alice unmasks $[\alpha]$ to obtain $[\beta_i] \leftarrow [c_{i-1}] \otimes [x_i]$ by distinguishing $R = 0$ and $R = 1$. If $R = 0$, $[\beta_i] \leftarrow [\alpha]$, else $[\beta_i] \leftarrow [1][\alpha]^{-1}$. Then, Alice computes $[c_i] \leftarrow [x_i]^{r_i}[c_{i-1}]^{r_i}[\beta_i]^{1-2r_i}$ and $[z_i] \leftarrow [x_i][r_i][c_{i-1}][c_i]^{-2}$. Afterward, Alice sets $i \leftarrow i+1$ and jumps to step 4 until $i = \ell - 1$. In order to get the equality testing result Alice computes $[\vartheta] \leftarrow \prod_{i=0}^{\ell-1}[1 - z_i]$ by running secure multiplication protocol as follows:
7: Alice chooses two random bits $r_i$ and $r_j$ and computes $[\alpha] \leftarrow [z_i \oplus r_i]$ and $[\beta] \leftarrow [z_j \oplus r_j]$. Then, Alice sends $[\alpha]$ and $[\beta]$ to Bob in packed form. Similar to step 4, using packing technique decreases number of decryption and communication cost significantly.
8: Bob decrypts and unpacks $[\alpha]$ and $[\beta]$, multiply them, and sends $[\theta] \leftarrow [\alpha \times \beta]$ to Alice.
9: Alice computes $[z_i \times z_j] \leftarrow [\theta][z_i]^{2r_ir_j - r_j}[z_j]^{2r_ir_j - r_i}[-r_ir_j]^{1/(1-2r_i-2r_j+4r_ir_j)}$.
```

the protocols in the semi-honest security model as long as the underlying cryptographic primitive is secure. Furthermore, In order to show that the protocols are privacy-preserving, we need to show Alice and Bob cannot learn new private information from each other. Recall that Alice only receives encrypted messages from Bob and she cannot distinguish ciphertexts, since the encryption scheme used in the protocols is semantically secure. Thus, it suffices to show that there is no information leakage to Bob in order to prove the improved equality testing protocols are privacy-preserving. It is clear that using packing technique does not leak any information since it uses homomorphic properties of the Paillier crypto-scheme. Therefore, we need to show that additive blindings used in the NEL-I are secure. In Steps 1, 3, and 5 of Protocol 3, Alice blinds her encrypted values additively before sending them to Bob. Thus, the decrypted messages in Bob are statistically indistinguishable from the original values before blinding. For blinding an ℓ-bit value a additively, Alice chooses a random value r that is κ bits longer than the actual a (κ = 80 bits), and then computes $a + r$. The security proof of additive blinding is related to statistical indistinguishability of $x = a + r$ from a random number x_R, which is drawn uniformly from $\{0, 1, \ldots, 2^{\ell+\kappa+1}\}$, as described in [9].

VI. Performance Analysis

In this section, we compare the performance of the EQT protocols based on analyzing computational complexities and the experimental results.

A. Computational complexity

Table II presents the computational complexities of the secure equality testing protocols in terms of multiplication and exponentiation. As an example, the exponentiation complexity of the ST06 protocol is $(6\ell)_{-1}$, which means there are 6ℓ exponentiations with a negative 1-bit exponents. It is clear that LT13 is the most expensive protocol because of the unmasking technique described in 4. To simplify the complexities and compare the protocols easier, we represent the complexities of exponentiation as multiplication. We can represent a ciphertext modulo n with an x-bit exponent as $3x/2$ multiplications modulo n. In Table II, overall complexity shows the complexity of each protocol represented as the number of multiplications. It can be observed that LT13 has a polynomial complexity, while ST06, NEL-I, and NEL-II are linear. Note that the complexities of encryption and decryption are not included in Table II since the protocols are crypto-scheme-independent and homomorphic crypto-schemes may have different encryption and decryption complexities.

Table II also presents the complexities of the secure exponentiation protocols used in LT13 and NEL-I, that are LT13(Expo) and NEL-I(Expo), respectively. Clearly, our new unmasking technique reduces the complexity of LT13(Expo) from $\mathcal{O}(n^\ell)$ to $\mathcal{O}(\ell^2 \log \ell)$. Note that the d value used in NEL-I(Expo) is between 0 and ℓ, resulting the exponential complexity to be $(\ell(\ell-1)/2)_{(\lceil \log_2 \ell/2 \rceil + \kappa)} + (2(\ell-1))_{-1}$. However, in NEL-I, the input of the secure exponentiation protocol is $\grave{d}$ that is between 0 and $\lceil \log_2 \ell \rceil$. Given the range of $\grave{d}$ in NEL-I, the complexity of the secure exponentiation protocol is $(\lceil \log_2 \ell \rceil(\lceil \log_2 \ell \rceil - 1)/2)_{(\lceil \log_2(\log_2 \ell/2) \rceil + \kappa)} + (2(\lceil \log_2 \ell \rceil - 1))_{-1}$, which is $\mathcal{O}((\log \ell)^2)$.

Table III presents the number of decryptions of the protocols, where ST06 has the highest number of decryptions. Table III shows that NEL-II has far fewer number of decryption compared to ST06, which results in a significantly much more efficient protocol when we consider the decryption cost.

TABLE III
Decryption complexities of the protocols.

Protocols	Decryption
LT13	$2N$
ST06	$N(3\ell - 1)$
NEL-I	$N(\ell + \kappa + 1)/n$
NEL-II	$\ell + N(\ell + \kappa + 4)/n + 2\lceil \log_2 \ell \rceil$

TABLE II
COMPUTATIONAL COMPLEXITIES OF THE SECURE EQUALITY TESTING AND THE EXPONENTIATION PROTOCOLS.

Protocols	Multiplication	Exponentiation	Overall complexity	
LT13	$(3/2)\ell + 3$	$1_{-1} + (\ell/2)_{-1} + 1_{(n/2)}$ $+ \sum_{i=2}^{\ell-1} 1_{(n/2)^i} + \ell_{(\ell/2)}$	$(1024)^{\ell-1} + 3\ell^2/4$ $+3\ell + 1540$	$\mathcal{O}(n^\ell)$
ST06	$11\ell + \ell/2 - 1$	$(6\ell)_{-1}$	$71\ell/2 - 1$	$\mathcal{O}(\ell)$
NEL-I	$1/2(\lceil \log_2 \ell \rceil(\lceil \log_2 \ell \rceil + 4) + \ell)$ $+3\rho$	$3\rho_{\ell+\kappa} + (\lceil \log_2 \ell \rceil(\lceil \log_2 \ell \rceil - 1)/2)_\kappa$ $+((\ell + \lceil \log_2 \ell \rceil)/2)_{-1}$	$\lceil \log_2 \ell \rceil(120\lceil \log_2 \ell \rceil - 115)$ $+7\ell/2 + 9366$	$\mathcal{O}(\ell)$
NEL-II	$11\ell + \ell/2 + 3\rho - 1$	$(6\ell)_{-1} + \rho_{\ell+\kappa} + (2\rho)_2$	$71\ell/2 + 3321$	$\mathcal{O}(\ell)$
LT13(Expo)	1	$1_{(n/2)} + \sum_{i=2}^{\ell-1} 1_{(n/2)^i}$	$(1042)^{\ell-1} + 1537$	$\mathcal{O}(n^\ell)$
NEL-I(Expo)	$(\ell^2 + 3\ell)/2 - 2$	$(\ell(\ell-1)/2)_{(\lceil \log_2 \ell/2 \rceil + \kappa)}$ $+(2(\ell-1))_{-1}$	$3/4\ell^2 \lceil \log_2 \ell/2 \rceil$ $+60\ell^2 - 55\ell - 5$	$\mathcal{O}(\ell^2 \log \ell)$

B. Experimental results

We implemented the protocols using C++ and external libraries: MPIR, Boost, and SeComLib on a single Linux machine running Ubuntu 14.04 LTS, with 64-bit microprocessor and 8 GB of RAM. The cryptographic key length of the Paillier is chosen according to NIST standards [20], which are valid until 2030. Table IV shows the parameters used in the implementation of the secure equality testing protocols. We analyze the performance of the protocols with different input sizes.

TABLE IV
PARAMETERS USED IN THE IMPLEMENTATION.

Parameter	Symbol	Value
Bit size of inputs	ℓ	2-30
Number of performed equality test	N	1000
Security parameter	κ	80 bits
Paillier message space	n	2048 bits

Figure 1 shows the run-time of the unmasking operations in LT13, NEL-I(a), and NEL-I. NEL-I(a) is another version of NEL-I, where the data packing is not used and the number of rounds is two (the degree of Lagrange polynomial is ℓ). As it is presented in Figure 1, NEL-I(a) is much more efficient than the LT13 due to the use of proposed secure exponentiation protocol. It also can be observed that adding one more round decreases the unmasking cost remarkably. Recall that adding one more round reduces the degree of polynomial from ℓ to $\lceil \log_2 \ell \rceil$. Consequently, number of unmasking operations are also decreased to $\lceil \log_2 \ell \rceil$, which makes the protocol computationally much more efficient.

Figure 2 presents the run-times of three different components of the protocols LT13, NEL-I, and NEL-I(a). These components are Paillier decryption, computation of $[\vartheta]$, and unmasking. Figure 2 shows that the unmasking cost of NEL-I(a) is significantly less than LT13. Figure 2 also shows that adding one more round and using data packing reduce the unmasking and decryption costs, respectively.

Figure 3 shows the run-times of the protocols. As expected, LT13 has the highest run-time among the others. It is clear that NEL-I has a much lower run-time compared to LT13, where the improvement is 99% for the 20-bit inputs. Figure 3 also shows a considerable improvement in NEL-II, where it's performance outperforms ST06 by 95% (for the 20-bit inputs) because of using data packing.

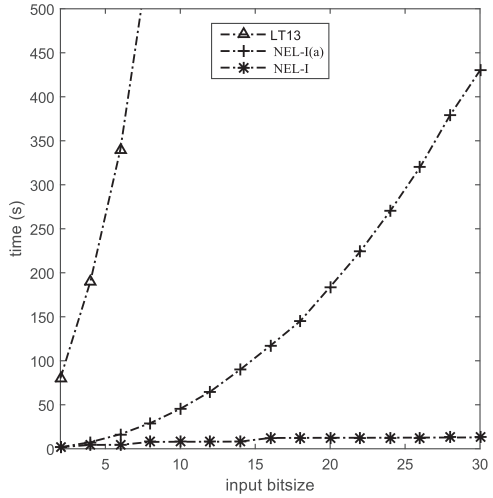

Fig. 1. Performance of the unmasking protocols.

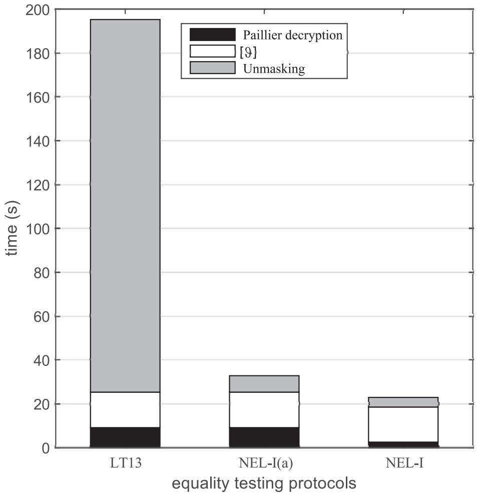

Fig. 2. Run-times of the protocols (4-bit inputs).

According to Figure 3, NEL-II outperforms slightly NEL-I in terms of computation. However, Table V shows ST06 and NELII both suffer from the high number of communication rounds because of bit-decomposition. Therefore, NEL-I is definitely a better choice for the applications with limited communicational resources.

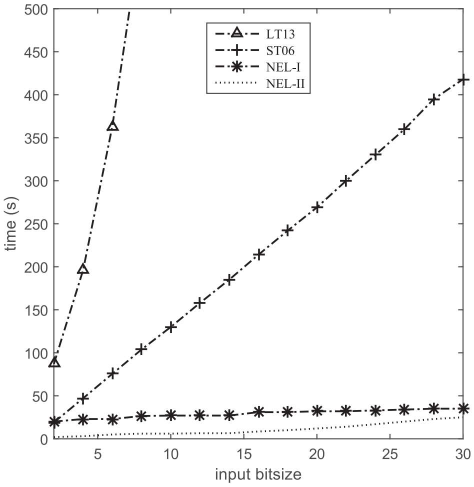

Fig. 3. Run-times of the protocols.

TABLE V
NUMBER OF COMMUNICATION ROUNDS OF THE PROTOCOLS.

Protocol	LT13	ST06	NEL-I	NEL-II
Round	2	$2\ell + 2\lceil \log_2 \ell \rceil + 2$	3	$2\ell + 2\lceil \log_2 \ell \rceil + 2$

VII. CONCLUSION

As one of the core building blocks, testing the equality of two encrypted integer values is very important. In this work, we describe the state-of-the-art cryptographic equality tests and propose two new protocols that are significantly more efficient. We achieve high performance by introducing algorithmic changes, an efficient exponentiation subroutine and deploying data packing. Experimental results show that our protocols are much more efficient in terms of computation than the existing works: achieving up to 99% improvement in run-time compared to the prior works in the field.

ACKNOWLEDGEMENTS

This publication is supported by the Dutch national program COMMIT.

REFERENCES

[1] A. Sadeghi, T. Schneider, and I. Wehrenberg, “Efficient privacy-preserving face recognition,” in *Information, Security and Cryptology - ICISC 2009, 12th International Conference, Seoul, Korea, December 2-4, 2009, Revised Selected Papers*, 2009, pp. 229–244.

[2] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, “Privacy-preserving face recognition,” in *Privacy Enhancing Technologies, 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5-7, 2009. Proceedings*, 2009, pp. 235–253.

[3] R. L. Lagendijk, Z. Erkin, and M. Barni, “Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 82–105, 2013.

[4] A. Jeckmans, A. Peter, and P. Hartel, “Efficient privacy-enhanced familiarity-based recommender system,” in *Computer Security–ESORICS 2013*. Springer, 2013, pp. 400–417.

[5] B. Pinkas, “Cryptographic techniques for privacy-preserving data mining,” *SIGKDD Explorations*, vol. 4, no. 2, pp. 12–19, 2002.

[6] Z. Yang, S. Zhong, and R. N. Wright, “Privacy-preserving classification of customer data without loss of accuracy,” in *Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21-23, 2005*, 2005, pp. 92–102.

[7] C. Li, R. Lu, H. Li, L. Chen, and J. Chen, “Pda: a privacy-preserving dual-functional aggregation scheme for smart grid communications,” *Security and Communication Networks*, vol. 8, no. 15, pp. 2494–2506, 2015.

[8] Z. Erkin and G. Tsudik, “Private computation of spatial and temporal power consumption with smart meters,” in *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings*, 2012, pp. 561–577.

[9] T. Veugen, “Encrypted integer division,” in *2010 IEEE International Workshop on Information Forensics and Security, WIFS 2010, Seattle, WA, USA, December 12-15, 2010*, 2010, pp. 1–6.

[10] M. Nateghizad, Z. Erkin, and R. L. Lagendijk, “An efficient privacy-preserving comparison protocol in smart metering systems,” *EURASIP J. Information Security*, vol. 2016, p. 11, 2016.

[11] B. Schoenmakers and P. Tuyls, “Efficient binary conversion for paillier encrypted values,” in *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, 2006, pp. 522–537.

[12] G. Couteau, “Efficient secure comparison protocols,” Cryptology ePrint Archive, Report 2016/544, 2016, http://eprint.iacr.org/2016/544.

[13] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in *2010 International Conference on Distributed Computing Systems, ICDCS 2010, Genova, Italy, June 21-25, 2010*, 2010, pp. 253–262.

[14] T. Nishide and K. Ohta, “Multiparty computation for interval, equality, and comparison without bit-decomposition protocol,” in *Public Key Cryptography–PKC 2007*. Springer, 2007, pp. 343–360.

[15] H. Lipmaa and T. Toft, “Secure equality and greater-than tests with sublinear online complexity,” in *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, 2013, pp. 645–656. [Online]. Available:

[16] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, 1999, pp. 223–238.

[17] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, “Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation,” in *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, 2006, pp. 285–304.

[18] J. Bar-Ilan and D. Beaver, “Non-cryptographic fault-tolerant computing in constant number of rounds of interaction,” in *Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing, Edmonton, Alberta, Canada, August 14-16, 1989*, 1989, pp. 201–209.

[19] T. Bianchi, A. Piva, and M. Barni, “Composite signal representation for fast and storage-efficient processing of encrypted signals,” *IEEE Trans. Information Forensics and Security*, vol. 5, no. 1, pp. 180–187, 2010.

[20] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, “Nist sp800-57: Recommendation for key management part 1: General (revised),” *NIST, Tech. Rep*, 2007.