HAL
open science

Benaloh's dense probabilistic encryption revisited

Laurent Fousse, Pascal Lafourcade, Mohamed Alnuaimi

To cite this version:

Laurent Fousse, Pascal Lafourcade, Mohamed Alnuaimi. Benaloh's dense probabilistic encryption revisited. AFRICACRYPT'11 - 4th international conference on Progress in cryptology in Africa, Jul 2011, Dakar, Senegal. pp.348-362. hal-00769449

HAL Id: hal-00769449

https://hal.science/hal-00769449v1

Submitted on 13 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Benaloh's Dense Probabilistic Encryption Revisited ⋆

Laurent Fousse[1], Pascal Lafourcade[2], and Mohamed Alnuaimi[3]

[1] Université Grenoble 1, CNRS, Laboratoire Jean Kuntzmann, France
Laurent.Fousse@imag.fr
[2] Université Grenoble 1, CNRS, Verimag, France
Pascal.Lafourcade@imag.fr
[3] Global Communication & Software Systems, United Arab Emirates
mohamed.alnuaimi@nkc.ae

Abstract. In 1994, Josh Benaloh proposed a probabilistic homomorphic encryption scheme, enhancing the poor expansion factor provided by Goldwasser and Micali's scheme. Since then, numerous papers have taken advantage of Benaloh's homomorphic encryption function, including voting schemes, private multi-party trust computation, non-interactive verifiable secret sharing, online poker. In this paper we show that the original description of the scheme is *incorrect*, because it can result in ambiguous decryption of ciphertexts. Then we show on several applications that a bad choice in the key generation phase of Benaloh's scheme has a real impact on the behaviour of the application. For instance in an e-voting protocol, it can inverse the result of an election. Our main contribution is a corrected description of the scheme (we provide a complete proof of correctness). Moreover we also compute the probability of failure of the original scheme. Finally we show how to formulate the security of the corrected scheme in a generic setting suitable for several homomorphic encryptions.

Keywords: public-key encryption, probabilistic encryption, homomorphic encryption scheme, Benaloh's scheme.

1 Introduction

An encryption scheme is homomorphic when it preserves some algebraic structure (usually group, sometimes ring) between the cleartext space and the ciphertext space, allowing computations on data encrypted with the same key. Examples of such encryptions are RSA [37] or ElGamal [19] which have the property that $\mathcal{E}(m_1) \times \mathcal{E}(m_2) = \mathcal{E}(m_1 \times m_2)$. In 1982 Goldwasser-Micali [25] introduced an encryption scheme with the different property $\mathcal{E}(b_1) \times \mathcal{E}(b_2) = \mathcal{E}(b_1 \oplus b_2)$. Several homomorphic encryption schemes have followed: Benaloh [3], Naccache and Stern [32], Okamoto and Uchiyama [33], Paillier [34] and its generalization

⋆ This work was supported by ANR SeSur AVOTE.

proposed by Damgård and Jurik [17], Sander, Young and Yung [40], Boneh et al [6]. All these schemes are partially homomorphic, meaning they allow homomorphic computation of only one operation (either addition or multiplication) on plaintexts. A cryptosystem allowing for homomorphic computation of two operations is called fully homomorphic. In 2009, Craig Gentry [21] found the first fully homomorphic encryption scheme, using lattice-based cryptography. However his scheme, while revolutionary, is not really practical and several recent works focus on concrete realizations of a fully homomorphic encryption scheme [41, 45, 22, 23]. Practitioners rely therefore on already existing partially homomorphic encryption. A survey of such cryptosystems can be found in [7] for non specialists, or in [2] with a complexity analysis. In [36], Rappe considers homomorphic cryptosystems and their applications, such as multiparty computation [12, 29, 18, 16], electronic voting [4, 9, 39, 38, 10, 3, 11, 15, 13, 28], key exchange using a server [44], non-interactive zero-knowledge [14], e-auction [1, 43, 8], non-interactive verifiable secret sharing [10], and others [27, 26, 20, 31].

Motivations and contributions: In 1994, Benaloh [3] proposed a homomorphic encryption which has a better expansion factor than Goldwasser-Micali's scheme [25]. This leads to a more practical scheme which has found several applications, such as voting schemes [4, 38, 10], private multi-party trust computation [12, 29, 18], non-interactive verifiable secret sharing [10], online poker [26]. Given all these applications of Benaloh's scheme, we were surprised to discover that its key generation process may in some cases lead to an ambiguous encryption.

Our first contribution is to show that the original scheme proposed by Benaloh in [3] does not give a unique decryption for all ciphertexts. We exhibit a simple example and characterize when this can happen and how to produce such counter-examples. The problem comes from the condition in public key generation: the original condition is not strong enough and allows to generate such keys that will compute ambiguous ciphertexts for some plaintexts.

Our second contribution is to describe how this error in key generation can have dramatic consequences in the applications of Benaloh's scheme. In each case we briefly explain how the application works on a simple example and show that a wrong key generation can have important consequences. In the case of the e-voting protocol it can change the result of an election; for private multi-party trust computation it can completely modify the computed trust value.

Our last contribution is a new condition (suitable for implementations) for the key generation which avoids such problems. We also compute the probability of failure of the original scheme, in order to understand why nobody discovered the problem before us. Moreover we discuss some schemes related to Benaloh's encryption. We also put the semantic security of the corrected encryption in the context of Kristian Gjøsteen's work [24]. Indeed revisited Benaloh's scheme can be seen as an instance of the general framework proposed for homomorphic cryptosystem based on subgroup membership problem.

Outline: In Section 2 we recall the original Benaloh scheme. In Section 3 we give a small example of parameters following the initial description and where we

have ambiguous decryption. In the next section, we discuss the (possibly serious) consequences of the problem we discovered in some applications. In Section 5 we give a corrected description of the scheme, with a proof of correctness. Then in Section 6, we analyze the probability of choosing incorrect parameters in the initial scheme. In Section 7 we discuss some schemes related to Benaloh's scheme. Finally before concluding, a semantic security analysis of the corrected scheme is given in Section 8.

2 Original Description of Benaloh's Scheme

Benaloh's "Dense Probabilistic Encryption" [3] describes a homomorphic encryption scheme with a significant improvement in terms of expansion factor compared to Goldwasser-Micali [25]. For the same security parameter (the size of the RSA modulus n), the ciphertext is in both cases an integer mod n, but the cleartext in Benaloh's scheme is an integer mod r for some parameter r depending on the key, whereas the cleartext in Goldwasser-Micali is only a bit. When computing the expansion factor for random keys, we found that it is most of the times close to 2 while it is $\lceil \log_2(n) \rceil$ for Goldwasser-Micali. We now recall the three steps of the original scheme given in Benaloh's paper [3].

Key Generation: The public and private key are generated as follows:

- Choose a block size r and two large primes p and q such that:
  - r divides $(p-1)$.
  - r and $(p-1)/r$ are relatively prime.
  - r and $q-1$ are relatively prime.
  - $n = pq$.
- Select $y \in (\mathbb{Z}_n)^* = \{x \in \mathbb{Z}_n : \gcd(x, n) = 1\}$ such that

$$y^{\varphi/r} \neq 1 \bmod n \tag{1}$$

  where φ denotes $(p-1)(q-1)$.

The public key is (y, r, n), and the private key is the two primes p and q.

Encryption: If m is an element in $\mathbb{Z}_r$ and u a random number in $(\mathbb{Z}_n)^*$ then we compute the randomized encryption of m using the following formula:

$$E_r(m) = \{y^m u^r \bmod n : u \in (\mathbb{Z}_n)^*\}.$$

It is easily verified that:

$$E_r(m_1) \times E_r(m_2) = E_r(m_1 + m_2).$$

Decryption: We first notice that for any m, u we have:

$$(y^m u^r)^{(p-1)(q-1)/r} = y^{m(p-1)(q-1)/r} u^{(p-1)(q-1)} = y^{m(p-1)(q-1)/r} \mod n.$$

Since $m < r$ and $y^{(p-1)(q-1)/r} \neq 1 \mod n$, Benaloh concludes that $m = 0 \bmod r$ if and only if $(y^m u^r)^{(p-1)(q-1)/r} = 1 \bmod n$. So if $z = y^m u^r \mod n$ is an encryption of m, given the secret key (p, q) we can determine whether $m = 0 \bmod r$. If r is small, we can decrypt z by doing an exhaustive search of the smallest non-negative integer m such that $(y^{-m} z \mod n) \in E_r(0)$. By precomputing values and using the baby-step giant-step algorithm it is possible to perform the decryption in time $O(\sqrt{r})$. Finally if r is smooth we can use classical index-calculus techniques. More details about these optimization of decryption are discussed in the original paper [3].

We remark that there is a balance to find between three parameters in this cryptosystem:

- ease of decryption, which requires that r is a product of small prime powers,
- a small expansion factor, defined as the ratio between the size of the ciphertexts and the size of the cleartexts. Because p and q have the same size and $r \mid p - 1$, this expansion factor is at least 2,
- strength of the private key, meaning that n should be hard to factorize. In the context of the P-1 factorization method [35], a large smooth factor of $p - 1$ is a definite weakness.

We notice that the cryptosystem proposed by Naccache-Stern [32] four years after Benaloh's scheme and based on the same approach addresses this issue and does not produce ambiguous encryption.

3 A Small Counter-Example

We start by picking a secret key $n = pq = 241 \times 179 = 43139$, for which we can set $r = 15$. Algorithm 1 may be used to compute the maximal suitable value of the r parameter if you start by picking p and q at random, but a smaller and smoother value may be used instead, for an easier decryption.

Algorithm 1 Compute r from p and q.

```
r ← p − 1;
while gcd(q − 1, r) ≠ 1 do
  r ← r/ gcd(r, q − 1);
end while
```

We verify that $r = 15$ divides $p - 1 = 240 = 16 \times 15$, r and $(p-1)/r = 16$ are relatively prime, $r = 15 = 3 \times 5$ and $q - 1 = 178 = 2 \times 89$ are coprime. Assume we pick $y = 27$, then $\gcd(y, n) = 1$ and $y^{(p-1)(q-1)/r} = 40097 \neq 1 \bmod n$ so

according to Benaloh's key generation procedure all the original conditions are satisfied.

By definition, $y^1 12^r = 24187 \bmod n$ is a valid encryption of $m_1 = 1$, while $y^6 4^r = 24187 \bmod n$ is also a valid encryption of $m_2 = 6$. In fact we can verify that with this choice of y, the true cleartext space is now $\mathbb{Z}_5$ instead of $\mathbb{Z}_{15}$ (hence the ambiguity in decryption): first notice that in $\mathbb{Z}_p$, $y^5 = 27^5 = 8 = 41^{15} = 41^r$. This means that a valid encryption of 5 is also a valid encryption of 0. For any message m, the set of encryptions of m is the same as the set of encryptions of $m+5$, hence the collapse in message space size. The fact that the message space size does not collapse further can be checked by brute force with this small set of parameters.

For this specific choice of p and q, there are $\frac{r-1}{r}\varphi(n) = 39872$ possible values of y according to the original paper, but 17088 of them would lead to an ambiguity in decryption (that's a ratio of 3/7), decreasing the cleartext space to either $\mathbb{Z}_3$ or $\mathbb{Z}_5$. Details are provided in Section 6.

4 Applications

In this section, we present some applications which explicitly use Benaloh's encryption scheme. We analyze the consequences of using a bad y parameter produced during the key generation for each application. In general, the ambiguity in the ciphertexts means that for a given cleartext $m \in \mathbb{Z}_r$ the value actually computed by the decryption algorithm is

$$m' = D(E_r(m)) \equiv m \bmod r' \tag{2}$$

with $r' \neq r$. Depending on the implementation of the discrete logarithm used by D (naive enumeration, baby steps/giant steps, possibly combined with a divide-and-conquer strategy when r is smooth) the value m' can be any of the values defined mod r that satisfy Equation (2). As the discrete logarithm algorithm used is not aware of the reduction from r to r', the impact in terms of computation time should be minimal (except for a naive increasing enumeration strategy which will always finish earlier, and where we are guaranteed to get the canonical solution mod r').

4.1 Receipt-free Elections

In [4], Benaloh and Tuinstra propose an application of homomorphic encryption for designing new receipt-free secret-ballot elections. They describe two protocols which use a homomorphic encryption scheme and verify a list of properties. They also give in the appendix of the paper a precise description of an encryption scheme which satisfies their properties. Its relation with [3] is given in Section 7.

The new voting protocol uses the fact that the encryption is homomorphic and probabilistic. If we have two candidates Nicolas and Ségolène then the system associates for instance the ballot 0 for Nicolas and the ballot 1 for Ségolène. The

main idea is that the server collects the m authenticated encrypted ballots $\{v_i\}_k$ corresponding to the choices v_i of the m voters. Then the server performs the multiplication of the ciphertexts to sum the votes and decrypts the product once to obtain the result. The number obtained corresponds to the number of votes n_S for Ségolène and the difference $m - n_S$ gives the number of votes for Nicolas.

We construct a basic application of the first protocol proposed in [4] and based on the example described in Section 3. In this example we consider only 12 voters. Suppose when the encryption is correctly done the final result is $\{11\}_k$. It means that after decryption Ségolène has 11 votes and Nicolas has 1 vote. But if as we explain in Section 3 instead of computing the result 11 mod 15 we are taking the result modulo 5, then we obtain a result of 11 mod $5 = 1$. This time the server concludes that Nicolas obtains 11 votes and Ségolène only 1. This example clearly shows that the flaw in the parameters generation process can have important consequences.

4.2 Private Multi-Party Trust Computation

In [18], Dolev *et al* give a multiple private key protocol for private multi-party computation of a trust value: an initiating user wants to know the (possibly weighted) average trust the network of nodes has in some user. In a first phase of the protocol, each of the n nodes splits its trust value t in $n-1$ shares (s_i) such that

$$t = s_1 + s_2 + \ldots + s_{n-1} \bmod r.$$

Here r is a common modulus chosen large enough with respect to the maximum possible global trust value, and in order to ensure the privacy of its trust value the shares should be taken as random number mod r, except for the last one. The shares are then sent encrypted (using Benaloh's scheme) to each other user, to be later recombined. If we assume that one of the users has chosen a faulty value for his public parameter y, then his contribution to the recombined value will be computed mod r' instead of mod r for some divisor r' of r. As an extreme example, assume

- that the queried user is a newcomer, untrusted by anyone (hence the private value of t for every node is 0),
- that the true recombined value contributed by the faulty user should have been $r - 1$,
- that $r' = r/3$.

Due to his miscalculation, the faulty node will contribute the value $r' - 1$ instead of -1, causing the apparent calculated trust value to be quite high (about 1/3 of the maximum possible trust value, instead of 0). This can have dramatic consequences if the trust value is used later on to grant access to some resource. These assumptions are not entirely unlikely: remember that $r = 3^k$ is an explicitly suggested choice of parameter of the cryptosystem in which we will find that the failure probability (ρ) is close to 1/3 and faulty nodes occur with high probability even with moderate-sized networks (see Section 6). We also note

that the description from [3] is given *in extenso*, with its incorrect condition. One reason for choosing Benaloh's cryptosystem in this application is because the cleartext space can be common among several private keys, a feature unfortunately not achieved *e.g.* by Paillier's cryptosystem [34] but also possible with Naccache-Stern's [32].

4.3 Privacy Preserving Clustering

In [29] the authors present another application of additive homomorphic encryption. They propose two new algorithms for solving the problem of privacy-preserving for clustering. Its allow them to solve the weighted average problem (WAP): two parties A and B knowing respectively (x, n) and (y, m) want to compute $\frac{x+y}{n+m}$ without revealing their own knowledge. The second algorithm they present uses a probabilistic semantically-secure additive homomorphic encryption scheme. They explicitly advise to use Benaloh's scheme. Their protocol works as follows:

1. A generates a Benaloh secret and public keys;
2. A knows (x, n) and B knows (y, m);
3. A starts the protocol by sending to B the two following encrypted messages $x_1 = E(x)$ and $n_1 = E(n)$.
4. B samples a random number z, computes $z_1 = E(z \times y)$ and $z_2 = E(z \times m)$. Moreover he sends to A the two following messages $m_1 = x_1^z \times z_1$ and $m_2 = n_1^z \times z_2$.
5. A decrypts m_1 and m_2 and performs the division $D(m_1)/D(m_2) = \frac{x+y}{n+m}$. Finally A sends to B the result of his computation.

A synthetic description of the protocol is given in Figure 1.

$$
\begin{array}{ccccc}
\text{A} & & & & \text{B} \\
(x, n) & \longrightarrow & x_1 = E(x) & \longrightarrow & \\
 & & n_1 = E(n) & & \\
 & & & & (y, m),\ z \leftarrow \mathit{Rand} \\
 & & & & z_1 = E(z \times y),\ z_2 = E(z \times m) \\
 & \longleftarrow & m_1 = x_1^z \times z_1 & \longleftarrow & \\
 & & m_2 = n_1^z \times z_2 & & \\
m_1 = E(z \times x + z \times y) & & & & \\
m_2 = E(z \times n + z \times m) & & & & \\
 & \longrightarrow & \frac{x+y}{n+m} & \longrightarrow &
\end{array}
$$

Fig. 1. Privacy preserving Weighted Average Protocol based on Benaloh' scheme E, introduced in [29]

A wrong choice in the key parameters could produce wrong values for m_1 and m_2, resulting in a wrong result shared by A and B as produced by the WAP

algorithm. As a consequence the privacy-preserving k-means algorithm would not compute the correct value. The authors have implemented their protocols using Benaloh's encryption and they provide execution timings (they claim that the implementation using Benaloh's encryption is more efficient that the one using oblivious polynomial evaluation). It is really surprising that they did not discover the problem, as they chose a value of $r = 3^k$ for which the probability of picking a faulty parameter is close to 1/3 (see §6).

4.4 Secure Cards Dealing

Another application of this encryption scheme is given in [26]: securely dealing cards in poker (or similar games). Here again the author gives the complete description of the original scheme, with a choice of parameter $r = 53$ (which is prime). Because r is prime, this application does not suffer from the flaw explained here, but this choice of a prime number is done for reasons purely internal to the cards dealing protocol, namely testing the equality of dealt cards.

Given two ciphertext $E(m_1)$ and $E(m_2)$, the players need to test if $m_1 = m_2$ without revealing anything more about the cards m_1 and m_2. The protocol is as follows:

1. Let $m = m_1 - m_2$, each player can compute $E(m) = E(m_1)/E(m_2)$ because of the homomorphic property of the encryption.
2. Each player P_i secretly picks a value $0 < \alpha_i < 53$, computes $E(m)^{\alpha_i}$ and discloses it to everyone.
3. Each player can compute $\prod_i E(m)^{\alpha_i} = E(m)^{\alpha}$ with $\alpha = \sum_i \alpha_i$. The players jointly decrypt $E(m)^{\alpha}$ to get the value $m\alpha \bmod r$.

Now because for each player the value of α is unknown and random, if $m\alpha \neq 0 \bmod r$ then the players learn nothing about m. Otherwise they conclude that the cards are equal.

We claim that this protocol fails to account for two problems:

- there is no guarantee that $\alpha \neq 0 \bmod r$. When this happens, two distinct cards will be incorrectly considered equal. One possible fix is to repeat the protocol to decrease the probability of false positive to an acceptable level.
- knowing the value of $E(m)$ and $E(m)^{\alpha_i}$, it is easy to recover α_i because of the small search space for α_i. This means the protocol leaks information when $m_1 \neq m_2$. The fix here is to multiply by some random encryption of 0.

It should be noted that these problems are unrelated to the incorrect parameter generation flaw discussed in this paper.

5 Corrected Version of Benaloh's Scheme

Let g be a generator of the group $(\mathbb{Z}_p)^*$, and since y is coprime with n, let α be the value in $\mathbb{Z}_{p-1}$ such that $y = g^{\alpha} \bmod p$. We will now state in Theorem 1 our main contribution:

Theorem 1 *The following properties are equivalent:*

a) α and r are coprime;
b) decryption works unambiguously;
c) for all prime factors s of r, we have $y^{(\varphi/s)} \neq 1 \bmod n$.

Of course property (b) is what we expect of the scheme, while (a) is useful to analyze the proportion of invalid y's and (c) is more efficient to verify in practice than (a), especially considering that in order to decrypt efficiently the factorization of r is assumed to be known. In the following proof we interpret statement (b) to mean that two different cleartexts cannot be encrypted to the same value:

$$\forall m_1, m_2 \in \mathbb{Z}_r, \forall u_1, u_2 \in (\mathbb{Z}_n)^*, \quad y^{m_1} u_1^r = y^{m_2} u_2^r \bmod n \Rightarrow m_1 = m_2 \bmod r.$$

Another way to interpret (b) is that, for a given $z \bmod n$, there is at most one value $m \bmod r$ such that $y^{-m} z$ is an r-th power mod n. In fact these two interpretations are equivalent: assume we can write

$$y^{-m_1} z = u_1^r$$
$$y^{-m_2} z = u_2^r$$

for two messages $m_1, m_2 \in \mathbb{Z}_r$ and two numbers $u_1, u_2 \in (\mathbb{Z}_n)^*$. Then

$$z = y^{m_1} u_1^r = y^{m_2} u_2^r$$

and the proof follows.

Proof. We prove first $(a) \Leftrightarrow (b)$ then we show $(a) \Leftrightarrow (c)$.

- We start by showing $(a) \Rightarrow (b)$. Assume two messages m_1 and m_2 are encrypted to the same element using nonce u_1 and u_2:

  $$y^{m_1} u_1^r = y^{m_2} u_2^r \bmod n.$$

  Reducing mod p we get:

  $$g^{\alpha(m_1 - m_2)} = (u_2/u_1)^r \bmod p$$

  and using the fact that g is a generator of $(\mathbb{Z}_p)^*$, there exists some β such that

  $$g^{\alpha(m_1 - m_2)} = g^{\beta r} \bmod p$$

  which in turns implies

  $$\alpha(m_1 - m_2) = \beta r \bmod p - 1.$$

  By construction r divides $(p-1)$, we can further reduce mod r and get

  $$\alpha(m_1 - m_2) = 0 \bmod r$$

  and since r and α are coprime, we can deduce $m_1 = m_2 \bmod r$, which means that decryption works unambiguously since the cleartexts are defined mod r.

– We now prove $(b) \Rightarrow (a)$. Assume α and r are not coprime and let $s = \gcd(\alpha, r)$, $r = sr'$, $\alpha = s\alpha'$. Then

$$\begin{aligned} y^{r'} &= g^{\alpha r'} \bmod p \\ &= (g^{\alpha'})^r \bmod p. \end{aligned}$$

Since r and $q-1$ are coprime, every invertible number mod q is an r-th power. Therefore $y^{r'}$ is an r-th power mod n and is a valid encryption of 0 as well as a valid encryption of r'.

– We now prove that $(a) \Rightarrow (c)$. Assume that there exists some prime factor s of r such that

$$y^{(\varphi/s)} = 1 \bmod n.$$

As above, by reducing mod p and using the generator g of $(\mathbb{Z}_p)^*$ we get

$$\alpha\frac{\varphi}{s} = 0 \bmod p-1.$$

So

$$\alpha\frac{\varphi}{s} = (p-1)\frac{\alpha(q-1)}{s}$$

is a multiple of $p-1$ and s divides $\alpha(q-1)$. Since s does not divide $q-1$, s divides α and α and r are not coprime.

– We now prove $(c) \Rightarrow (a)$. Assume α and r are not coprime and denote by s some common prime factor. Then

$$\begin{aligned} y^{(\varphi/s)} &= g^{\alpha\varphi/s} \bmod p \\ &= g^{(\alpha/s)\varphi} \bmod p = 1 \bmod p. \end{aligned}$$

And by construction of r, $s \nmid q-1$ so $y^{(\varphi/s)} = 1 \bmod q$. □

Notice than in the example of Section 3 we have $y^{(p-1)(q-1)/3} = 1 \bmod n$ so condition (c) is not satisfied. We claimed that the real ciphertext space is now $\mathbb{Z}_5$, and we gave a precise analysis of the cleartext space reduction at the end of Section 6.

6 Probability of Failure of Benaloh's Scheme

We now estimate the probability of failure in the scheme as originally described. For this we need to count the numbers y that satisfy Equation (1) in Section 2 and not property (c) of Theorem 1. We call these values of y "faulty".

Lemma 1 *Equation (1) is equivalent to the statement:* $r \nmid \alpha$.

Proof. Assume that r divides α: $\alpha = r\alpha'$. So

$$\begin{aligned} y^{\varphi/r} &= g^{\alpha\varphi/r} \bmod p \\ &= (g^{\alpha'})^{\varphi} \bmod p \\ &= 1 \bmod p. \end{aligned}$$

Since r divides $p-1$, $y^{\varphi/r} = 1 \bmod\ q$ hence $y^{\varphi/r} = 1 \bmod\ n$.

Conversely, if $y^{\varphi/r} = 1 \bmod n$, then

$$\begin{aligned} g^{\alpha\varphi/r} &= 1 \bmod p \\ \alpha\frac{\varphi}{r} &= 0 \bmod p-1. \end{aligned}$$

Since r divides $p-1$ and is coprime with $\frac{\varphi}{r}$ (by definition), we have $r \mid \alpha$. $\square$

Since picking $y \in (\mathbb{Z}_p)^*$ at random is the same when seen mod p as picking $\alpha \in \{0, \ldots, p-2\}$ at random, we can therefore conclude that the proportion ρ of faulty y's is exactly the proportion of non-invertible numbers mod r among the non-zero mod r. So $\rho = 1 - \frac{\varphi(r)}{r-1}$. We notice that this proportion depends on r only, and it is non-zero when r is not a prime. Since decryption in Benaloh's scheme is essentially solving a discrete logarithm in the subgroup of $(\mathbb{Z}_p)^*$ of order r, the original scheme recommends to use r as a product of small primes' powers, which tends to increase ρ. In fact, denoting by (p_i) the prime divisors of r we have:

$$\begin{aligned} \rho &= 1 - \frac{\varphi(r)}{r-1} \\ &= 1 - \frac{r}{r-1}\frac{\varphi(r)}{r} \\ &= 1 - \frac{r}{r-1}\prod_i \frac{p_i - 1}{p_i} \\ &\approx 1 - \prod_i \frac{p_i - 1}{p_i} \end{aligned}$$

which shows that the situation where decryption is easy also increases the proportion of invalid y's when using the initial description of the encryption scheme.

As a practical example, assume we pick two 512 bits primes p and q as

$$\begin{aligned} p &= 2 \times (3 \times 5 \times 7 \times 11 \times 13) \times p' + 1 \\ p' &= 4464804505475390309548459872862419622870251688508955\backslash \\ &\quad 5037374496982090456310601222033972275385171173585381\backslash \\ &\quad 3914691524677018107022404660225439441679953592 \\ q &= 1005585594745694782468051874865438459560952436544429\backslash \\ &\quad 5033292671082791323022555160232601405723625177570767\backslash \\ &\quad 5238936398645381403154121089599274598252367545682 79. \end{aligned}$$

Then

$$\begin{aligned}
\gcd(q-1, p-1) &= 2 \\
r &= (3 \times 5 \times 7 \times 11 \times 13) \times p' \\
\rho &= 1 - \frac{r}{r-1} \times \frac{2}{3} \times \frac{4}{5} \times \frac{6}{7} \times \frac{10}{11} \times \frac{12}{13} \times \frac{p'-1}{p'} \\
\rho &> 61\%.
\end{aligned}$$

This example was constructed quite easily: first we take p' of suitable size, and multiply its value until $p = k \times p' + 1$ is prime. Then we generate random primes q of suitable size until the condition $\gcd(p-1, q-1) = 2$ is verified; it took less than a second on a current laptop using Sage [42].

Putting it all together, we can also characterize the faulty values of y, together with the actual value r' of the cleartext space size (compared to the expected value r):

Lemma 2 *Let $u = \gcd(\alpha, r)$. Then $r' = \frac{r}{u}$. Moreover if $r' \neq r$, this faulty value of y goes undetected by the initial condition as long as $u \neq r$.*

Proof. Let $\hat{r} = \frac{r}{u}$ and $\alpha' = \frac{\alpha}{u}$. Consider

$$\begin{aligned}
y^{\hat{r}} &= g^{u\hat{r}\alpha'} \bmod p \\
&= (g^{\alpha'})^r \bmod p.
\end{aligned}$$

Since r is coprime with $q-1$, $y^{\hat{r}}$ is an r-th power mod q. Hence $y^{\hat{r}}$ is a valid encryption of $\hat{r}$ and of 0 at the same time, which means $r'|\hat{r}$.

We need to prove that the smallest positive power of y which is an r-th power is $y^{r'}$. Assume

$$y^m = u^r$$

for some $u \in (\mathbb{Z}_n)^*$. Then

$$\begin{aligned}
y^m &= u^r \bmod n \\
g^{\alpha m} &= u^r \bmod p \\
&= g^{\beta r} \bmod p \text{ for some } \beta \\
\alpha m &= \beta r \bmod p-1 \\
\alpha m &= 0 \bmod r \\
\alpha' u m &= 0 \bmod r \\
\alpha' m &= 0 \bmod \hat{r} \\
m &= 0 \bmod \hat{r}
\end{aligned}$$

which proves that the effective cleartext space size r' is at least $\hat{r}$.

The second point of Lemma 2 is a mere rephrasing of the previous lemma.

This result can be used to craft counter-examples as we did in Section 3: for a valid value y of the parameter and u a proper divisor of r, the value $y' = y^u \bmod n$ is an undetected faulty value with actual cleartext space size $r' = r/u$. It can also be used to determine precisely, for every proper divisor r' of r the probability of picking an undetected faulty parameter y of actual cleartext space size r'. Such an extensive study was not deemed necessary in the examples of Section 4, but it confirms that ambiguous parameters can happen more frequently than expected.

7 Related Schemes

We briefly discuss in this section some schemes related to that of [3].

In [4], Benaloh and Tuinstra describe a cryptosystem which closely resembles that of [3], but the conditions given on r are less strict. Let us recall briefly the parameters of the cryptosystem as described in [4]:

- $r \mid p-1$ but $r^2 \nmid p-1$.
- $r \nmid q-1$.
- y is coprime with n and $y^{(p-1)(q-1)/r} \neq 1 \bmod n$.

It is clear that $r^2 \nmid p-1$ is weaker than $\gcd((p-1)/r, r) = 1$, and that $r \nmid q-1$ is weaker than $\gcd(q-1, r) = 1$. Therefore any set of parameters satisfying [3] are also valid parameters as defined in [4].

Unfortunately the condition imposed on y is the same and still insufficient, and finding counter-examples is again a matter of picking α not coprime with r. Our theorem still stands for this cryptosystem if you replace condition (c) by the following condition:

$$\text{For all prime factors } s \text{ of } r, \text{ we have } y^{(p-1)/s} \neq 1 \bmod p. \tag{3}$$

Going back in time, the scheme of Goldwasser and Micali [25] can be seen as a precursor of [4] with a fixed choice of $r = 2$. The choice of y in [25] as a quadratic non-residue mod n is clearly an equivalent formulation of condition (3).

Before [3] and [4], the scheme was defined by Benaloh in [5], with the parameter r being a prime. In this case our condition (c) is the same as the one proposed by Benaloh, and the scheme in this thesis is indeed correct. The main difference between the different versions proposed afterwards and this one is that it is not required for r to be prime, which leads in some cases to ambiguous ciphers. This remark clearly shows that all details are important in cryptography and that the problem we discovered is subtle because even Benaloh himself did not notice it.

Finally the scheme proposed by Naccache and Stern [32] is quite close to the one proposed in [5] but with a parameterization of p and q. It makes decryption correct, efficient, and leaves the expansion factor as an explicit function of the desired security level with respect to methods of factoring taking advantage of this specific form of n, like the $P-1$ method [35] (the expansion is essentially the

added size of the big cofactors of $p-1$ and $q-1$). If we drop this requirement that $p-1$ and $q-1$ have big cofactors, their scheme becomes a corrected generalization of Benaloh's, so application writers should probably use Naccache-Stern's scheme directly. We note that a modulus size of 768 bits was considered secure at the time, a fact disproved twelve years later [30] only!

8 Semantic Security of the Corrected Scheme

In [24], Kristian Gjøsteen formulates the security of several homomorphic encryption schemes in a common setting and relates the semantic security of the schemes to a generic problem (the Decisional Subgroup Membership Problem) which we recall here:

Problem 1 (DSMP). Let G be an abelian group with subgroups K, H such that $G = KH$ and $K \cap H = \{1\}$. The *Decisional Subgroup Membership Problem* is to decide whether a given $g \in G$ is in K or not.

The cryptosystems by Goldwasser-Micali, Naccache-Stern, Okamoto-Uchiyama and Paillier respectively are shown to fit in this setting, with a proper definition for G (the ciphertexts space), H (coding the cleartexts) and K (the "cloak" space used to randomize encryptions). For example for Paillier's encryption, the ciphertext space is $G = (\mathbb{Z}_{n^2})^* \simeq (\mathbb{Z}_n)^* \times \mathbb{Z}_n$, the cleartexts coding subgroup H is the subgroup of order n (generated by $g = 1 + n$) and K is the set of the invertible n-th powers mod n^2. This is consistent with the probabilistic encryption function

$$E_u(m) = (1 + n)^m u^n \bmod n^2.$$

It can be verified quite easily that the following choices make the corrected version of Benaloh's scheme fit in this setting:

- $G = (\mathbb{Z}_n)^*$
- H the cyclic subgroup of order r of G
- K the set of invertible r-th powers in G
- the public element y must generate H.

Using the result in [24], the semantic security of our corrected scheme is therefore equivalent to the DSMP for K, that is, being able to distinguish r-th powers modulo n.

Although several homomorphic encryption schemes are analyzed in [24], Benaloh's is not. Our correction ensures that the last condition is met, otherwise y could generate a strict subgroup of the intended group H.

9 Conclusion

We have shown that the original definition of Benaloh's homomorphic encryption does not give sufficient conditions in the choice of public key to get an

unambiguous encryption scheme. We also explain on some examples what can be the consequences of the use of the original Benaloh scheme. Our discussion on the probability of choosing an incorrect public key shows that this probability is non negligible for parameters where decryption is efficient: for example using the suggested value of the form $r = 3^k$, this probability is already close to 1/3. Our main contribution is to propose a necessary and sufficient condition which fixes the scheme. In fact, it is surprising this result was not found before, considering the number of applications built on the homomorphic property of Benaloh's scheme. This strongly suggests this scheme was rarely implemented or even worse, implementations were rarely well tested.

References

1. Masayuki Abe and Koutarou Suzuki. M+1-st price auction using homomorphic encryption. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*, volume 2274 of *Lecture Notes in Computer Science*, pages 115–124. Springer, 2002.
2. M. Akinwande. Advances in Homomorphic Cryptosystems. *Journal of Universal Computer Science*, 15(3):506–522, 2009.
3. Josh Benaloh. Dense Probabilistic Encryption. In *In Proceedings of the Workshop on Selected Areas of Cryptography*, pages 120–128, 1994.
4. Josh Benaloh and Dwight Tuinstra. Receipt-free Secret-Ballot Elections (extended abstract). In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 544–553, New York, NY, USA, 1994. ACM.
5. Josh Daniel Cohen Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, New Haven, CT, USA, 1987.
6. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.
7. Caroline Fontaine and Fabien Galand. A Survey of Homomorphic Encryption for Nonspecialists. In *EURASIP Journal on Information Security*. Hindawi Publishing Corporation, 2007.
8. Xiaofeng Chen, Byoungcheon Lee, and Kwangjo Kim. Receipt-free electronic auction schemes using homomorphic encryption. In Jong In Lim and Dong Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers*, volume 2971 of *Lecture Notes in Computer Science*, pages 259–273. Springer, 2003.
9. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *26th Annual Symposium on Foundations of Computer Science, 21-23 October 1985, Portland, Oregon, USA*, pages 372–382. IEEE, 1985.
10. Josh Cohen Benaloh. Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret. In *Proceedings on Advances in Cryptology—CRYPTO '86*, pages 251–260, London, UK, 1987. Springer-Verlag.

11. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'97)*, volume 1233, pages 103–118, Konstanz, Germany, 1997. Springer-Verlag.
12. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty Computation from Threshold Homomorphic Encryption. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 280–299, London, UK, 2001. Springer-Verlag.
13. Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. Multi-autority secret-ballot elections with linear work. In *EUROCRYPT*, pages 72–83, 1996.
14. Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 41–59. Springer, 2006.
15. Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen. A generalization of paillier's public-key system with applications to electronic voting. *Int. J. Inf. Sec.*, 9(6):371–385, 2010.
16. Ivan Damgård and Jesper Buus Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 247–264. Springer, 2003.
17. I. Damgård and M. Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *Public Key Cryptography*, page 119–136, 2001.
18. Shlomi Dolev, Niv Gilboa, and Marina Kopeetsky. Computing Multi-Party Trust Privately: in O(n) time units sending one (possibly large) message at a time. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1460–1465, New York, NY, USA, 2010. ACM.
19. Taher Elgamal. Proceedings of CRYPTO 84 on Advances in Cryptology. In *A Public key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, pages 10–18. Springer-Verlag New York, Inc., Santa Barbara, California, United States, 1985.
20. Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.
21. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.
22. Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 116–137. Springer, 2010.

23. Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. -hop homomorphic encryption and rerandomizable yao circuits. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 155–172. Springer, 2010.
24. Kristian Gjøsteen. Homomorphic cryptosystems based on subgroup membership problems. In Ed Dawson and Serge Vaudenay, editors, *Progress in Cryptology – Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 314–327. Springer Berlin / Heidelberg, 2005.
25. Shafi Goldwasser and Silvio Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *STOC*, pages 365–377, 1982.
26. Philippe Golle. Dealing Cards in Poker Games. In *Proc. of ITCC 2005 E-Gaming Track*, 2005.
27. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. *J. Cryptology*, 23(4):546–579, 2010.
28. Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556. Springer, 2000.
29. Somesh Jha, Luis Kruger, and Patrick McDaniel. Privacy Preserving Clustering. In Sabrina de Capitani di Vimercati, Paul Syverson, and Dieter Gollmann, editors, *Computer Security – ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 397–417. Springer Berlin / Heidelberg, 2005.
30. Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Pierrick Gaudry, Peter L. Montgomery, Dag Arne Osvik, Herman Te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit RSA modulus, 2010.
31. Helger Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In Chi-Sung Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 416–433. Springer, 2003.
32. David Naccache and Jacques Stern. A New Public Key Cryptosystem Based on Higher Residues. In *ACM Conference on Computer and Communications Security*, pages 59–66, 1998.
33. T. Okamoto and S. Uchiyama. A New Public-key Cryptosystem as Secure as Factoring. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'98)*, volume 1403, pages 308–318, Helsinki, Finland, 1998. Springer-Verlag. Lecture Notes in Computer Science.
34. P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'99)*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, 1999. Springer-Verlag.
35. J. M. Pollard. Theorems on Factorization and Primality Testing. *Mathematical Proceedings of the Cambridge Philosophical Society*, 76(03):521–528, 1974.
36. Doerte K. Rappe. Homomorphic Cryptosystems and their Applications. Cryptology ePrint Archive, Report 2006/001, 2006. http://eprint.iacr.org/.
37. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, February 1978.

38. Alexandre Ruiz and Jorge Luis Villar. Publicly Verifiable Secret Sharing from Paillier's Cryptosystem. In Christopher Wolf, Stefan Lucks, and Po-Wah Yau, editors, *WEWoRC*, volume 74 of *LNI*, pages 98–108. GI, 2005.
39. Kazue Sako and Joe Kilian. Secure voting using partially compatible homomorphisms. volume 839 of *Lecture Notes in Computer Science*, pages 411–424. Springer, 1994.
40. Tomas Sander, Adam Young, and Moti Yung. Non-Interactive CryptoComputing for NC^1. In *FOCS*, pages 554–567, 1999.
41. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.
42. W. A. Stein et al. *Sage Mathematics Software (Version 4.5.1)*. The Sage Development Team, 2010. `http://www.sagemath.org`.
43. Koutarou Suzuki and Makoto Yokoo. Secure generalized vickrey auction using homomorphic encryption. In Rebecca N. Wright, editor, *Financial Cryptography, 7th International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003, Revised Papers*, volume 2742 of *Lecture Notes in Computer Science*, pages 239–249. Springer, 2003.
44. Makoto Tatebayashi, Natsume Matsuzaki, and David B. Newman. Key distribution protocol for digital mobile communication systems. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)*, volume 435, pages 324–333, Santa Barbara, California, USA, 1989. Springer-Verlag.
45. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.