Short paper: An Update on Marked Mix-Nets: An Attack, A Fix and PQ Possibilities

Thomas Haines[1], Olivier Pereira[2], and Peter Roenne[3]

[1] Norwegian University of Science and Technology, Trondheim, Norway
thomas.haines@ntnu.no
[2] Université catholique de Louvain, Louvain, Belgium
olivier.pereira@uclouvain.be
[3] Université du Luxembourg, Luxembourg, Luxembourg
peter.roenne@uni.lu

Abstract. Marked mix-nets were introduced by Pereira and Rivest as a mechanism to allow very efficient mixing that ensures privacy but at the cost of not guaranteeing integrity. This is useful in a number of e-voting schemes such as STAR-Vote and Selene. However, the proposed marked mix-net construction comes with no proof of security and, as we show in this paper, does not provide privacy even in the presence of a single corrupt authority. Fortunately, the attack that we present is easy to prevent and we show several possible ways to address it. Finally while the original marked mix-net paper worked with ElGamal, we identify conditions that the adopted encryption scheme should satisfy in order to be appropriate for a marked mix-net. This opens the possibility of building marked mix-nets based on intractability assumptions which are believed to hold in the presence of a quantum computer.

1 Introduction

Marked mixnets [9] are a technique proposed by Pereira and Rivest to enable faster mixing by only restricting attacks on privacy but not integrity attacks. At first it may seem strange to even consider a mix-net which only provides privacy but not integrity. However, in a variety of applications, we can (and sometimes must) independently check the output of a mix-net for correctness, and it then suffices to have privacy. This is notable the case in e-voting which schemes like STAR-vote [3] and Selene [11]. Another possible use case is to produce election results fast for public elections where tally time is often critical, and postpone the verifiability proofs until after the election result. In this case, the marked mixnet process constitute an accountable commitment to the result from each mixer server's side, and already offers some verifiable privacy guarantees (compared to a solution in which each mixer would simply shuffle ciphertexts).

The main idea of the marked mix-net is to have a sequence of mixing nodes that shuffle and reencrypt ciphertexts (as usual in any reencryption mixnet), with the twist that each mix node also a secret mark on its output ciphertexts. This mark prevents a later mix node from bypassing one or more earlier mixers

by using their input ciphertexts. On top of this, voters are required to include a random value in each of their ciphertext, and to make each ciphertext somehow non-malleable, so that ciphertext copies (which could be a threat to privacy) can be identified at decryption time. This identification is expected to be a sufficent deterrent for cheating mixers (this is often called the *covert adversary* model [1]).

At present there is no rigorous security definition, and even less proof of correctness for the marked mix-net technique and, in this paper, we indeed present an attack. The essence of the attack is to exploit the homomorphic properties of ElGamal used in prime order groups to circumvent the marking mechanism, hence making it possible for the last mixer to bypass the earlier mixers and completely break privacy. We elaborate on the attack in Section 3.

We present two options that make it possible to prevent our attack: one is generic, and requires each mixer to add an extra dummy ciphertext as part of their mixing process. This should restore security as long as no adversary can guess which ciphertext is dummy with overwhelming probability. Our second option is to perform ElGamal encryption in a group of unknown order: it is not generic anymore, but keeps the marked mix-net protocol unchanged. We elaborate on this in Section 4.

Finally, whereas the original marked mix-net construction is based on a cryptosystem relying on classical hardness problems, we suggest that it would be possible to apply that construction to any publicly rerandomizable RCCA cryptosystem [5] meeting some minor additional constrains. This includes possible cryptosystems built on lattice based assumptions which are believed to hold even in the presence of large scale quantum computers. In the classical setting marked mixnets are only a small factor faster overall, though they are faster in the online phase by a factor of at least 100 compared to fully verifiable mixnets.[4] In the post-quantum setting, this efficiency gain is likely to be much higher.[5] Since we generalise a scheme with no rigorous security definition or proof of correctness, we claim only that our generalisation does not break the completeness of marked mix-nets nor does it invalidate any of the security arguments presented in the original paper.

2 Marked Mix-Net Construction

The original paper describes the scheme for the specific case of ElGamal encryption of OAEP3 transformed messages [10]. In this work we generalise this to (publicly) randomizable RCCA secure encryption schemes [5]; we make two additional requirements on the scheme but these appears to be hold for most currently known instantiations.

[4] This speed-up occurs because every verifiable mix requires at least one online exponentiation per ciphertext, while the marked mix-net only requires one online ciphertext multiplication per ciphertext.

[5] To our knowledge there is no published post-quantum verifiable mixnet with clear benchmarks and hence providing a concrete efficiency comparison is left as an interesting open problem.

2.1 Primitives

Definition 1. *Rerandomisable Public Key Encryption scheme (Rand-PKE). A re-randomisable PKE is a tuple of five algorithms (Setup, KGen, Enc, Dec, Rand).*

- $Setup(1^\lambda)$ *on input the security parameter* λ *outputs the public parameters* prm.
- $KGen(prm)$ *on input the public parameters* prm, *outputs a key pair* (pk, sk).
- $Enc(pk, M)$ *on input a public key* pk *and message* M *outputs a ciphertext* C.
- $Dec(sk, C)$ *on input a public key* pk, *correponding secret key* sk, *and ciphertext* C, *outputs a message* M *or error symbol* $\perp$.
- $Rand(pk, C)$ *on input a public key* pk *and ciphertext* C, *outputs another ciphertext* C'.

Definition 2. *Rand-PKE correctness, We say a PKE scheme* $\mathcal{PKE}$ *is correct if*

$$\forall \lambda, prm \leftarrow Setup(1^\lambda), (pk, sk) \leftarrow KGen(prm), \forall M \ , \\ Dec(sk, Enc(pk, M)) = M \ .$$

We will now define the security properties for Rand-PKE. The first definition, Def. 3, is the standard definition of indistinguishability for Rand-PKE from Canetti et al.'s original paper [5]. The adversary chooses two messages based on the public key and with access to the decryption oracle $Dec^\diamond$. A ciphertext C is then created for the message M_b. The adversary must guess if b is equal to 0 or 1, it does this with the state st it gave when it created the messages, the ciphertext C and with access to the decryption oracle which will decrypt any ciphertext which does not decrypt to either M_0 or M_1.

The second definition, Def. 4, from Groth [8] captures the ability of an adversary who constructed the ciphertext to tell the difference between a re-encryption of the ciphertext or a new fresh encryption of the same message; the definition we use is weak in the sense that it does not capture an adversary that knows the private key material. Marked mixnets provide privacy under the first definition but cannot provide receipt-freeness without the second.

Definition 3 (Replayable CCA Security, [5]). *Consider the experiment in Fig 1. We say a PKE scheme* $\mathcal{PKE}$ *is indistinguishable secure under replayable chosen-ciphertext attacks (RCCA-secure) if for all PPT adversaries* $\mathcal{A}$*:*

$$\boldsymbol{Adv}^{RCCA}_{\mathcal{A},\mathcal{PKE}}(\lambda) := \left| Pr[\boldsymbol{Exp}^{RCCA}_{\mathcal{A},\mathcal{PKE}}(\lambda) = 1] - \frac{1}{2} \right| \in \boldsymbol{negl}(\lambda).$$

Definition 4 (Weak Rerandomisabilty [8]). *Consider the experiment in Fig. 2. Let* $\mathcal{PKE}$ *be a re-randomisable PKE scheme.* $\mathcal{PKE}$ *is rerandomizable*

$\mathbf{Exp}^{RCCA}_{\mathcal{A},\mathcal{PKE}}(\lambda)$

$prm \leftarrow Setup(1^\lambda), b \leftarrow_r \{0,1\}$;
$(pk, sk) \leftarrow KGen(prm)$;
$(M_0, M_1, st) \leftarrow A^{Dec(sk,\cdot)}(pk)$;
$C \leftarrow Enc(pk, M_b)$;
$b' \leftarrow A^{\mathrm{Dec}^\diamond(sk,\cdot)}(st, C)$;
$return(b' = b^*)$.

$Dec^\diamond(sk, \cdot)$

Upon input C;
$M' \leftarrow \mathrm{Dec}(sk, C)$;
if $M' \in \{M_0, M_1\}$ then output $\diamond$,
else output M'.

Fig. 1. The RCCA Security Experiment

$Exp^{Rand-wRCCA}_{\mathcal{A},\mathcal{PKE}}(\lambda)$

$prm \leftarrow Setup(1^\lambda), b \leftarrow_r \{0,1\}$;
$(pk, sk) \leftarrow KGen(prm)$;
$C \leftarrow A^{Dec(sk,\cdot)}(pk)$;
$M \leftarrow \mathrm{Dec}(sk, C)$;
if $M = \perp$ return b;
if $b = 0$ then $C^* \leftarrow Enc(pk, M)$,
else $C^* \leftarrow Rand(pk, C)$;
$b' \leftarrow A^{Dec^\perp(sk,\cdot)}(pk, C^*)$;
$return(b' = b^*)$.

$Dec^\perp(sk, \cdot)$

Upon input C;
$M' \leftarrow Dec(sk, C)$;
if $M' = M$ then output $\perp$,
else output M'.

Fig. 2. Weak Re-randomizable RCCA encryption

under weak replayable chosen-ciphertext attacks (Rand-wRCCA secure) if for all PPT adversaries $\mathcal{A}$:

$$\left| Pr[\boldsymbol{Exp}^{Rand-wRCCA}_{\mathcal{A},\mathcal{PKE}}(\lambda) = 1] - \frac{1}{2} \right| \in \boldsymbol{negl}(\lambda).$$

In addition to the standard properties of RCCA schemes, we additionally require that with knowledge of the secret key material it is possible to check if two ciphertexts are re-encryptions of the same original ciphertext (we note that this rules out anything stronger than Weak-RCCA Re-randomisabilty and hence receipt freeness in regards to the authorities is not possible to achieve if the input ciphertexts are directly linked to the voters.).

Marked mixnets, further, require that some subsection of the RCCA ciphertext space forms a homomorphic IND-CPA scheme. We note that this is almost always true since RCCA schemes are constructed from IND-CPA by adding a transform, a hash, or a signature.

Finally we assume that all the methods can be efficiently distributed among the authorities in a threshold way, which is also true of most RCCA schemes. Some examples of RCCA secure encryption schemes with Weak-RCCA are OAEP3 transformed ElGamal [10], the scheme of Faonio and Fiore [7], and the post quantum construction of [2].

We leave a formal definition of these requirements as future work.

2.2 Construction

We present the construction in a slightly simplified form; for most RCCA schemes it is possible to compute ahead of time most of **Rand** before seeing the particular ciphertext being re-randomised, this should be done in the setup phase. We refer the reader to the original paper [9] for the description in the concrete case of OAEP3 transformed ElGamal.

Setup The authorities jointly run $Setup(1^\lambda)$ to produce the public parameters prm. They then run $KGen(prm)$ to produce the keys pk and sk. pk is published to the bulletin board.

Submission Each sender encrypts their input M_i by running $Enc(pk, M_i)$ and receiving C_i. They then post their ciphertext to the bulletin board.

Mixing In the mixing phase, each mixer chooses a single mark a_i—from the message space of RCCA scheme, and posts $Enc(pk, a_i)$ to the bulletin board. They then permute the list of inputs (or the output of the previous mix server), rerandomize them using $Rand$, and adds—in the case of OAEP3 transformed ElGamal multiplies—the plaintext mark to the homomorphic space.

Decryption Once mixing is over, the authorities decrypts all the marks, which are then homomorphically removed from the ciphertexts. The authorities then check that the ciphertexts are valid and are independent.

Intuitively, it is expected that this mixing processes guarantees that all the decrypted ciphertexts have being rerandomized and shuffled by all the mixers, since the mark of every mixer appears on each ciphertext at decryption time. No correctness guarantee is offered, though: the first mixer, for instance, is perfectly free to mix whatever list of ciphertexts he likes, independently of its expected inputs.

3 Attack

For simplicity we present both the attack and the fix for the concrete RCCA scheme of OAEP3 transformed ElGamal which was suggested in the original marked mix-nets paper. We do not enter into the details of OAEP3-ElGamal encryption: for our purpose, it is sufficient to know that it is identical to the traditional ElGamal encryption algorithm, except that messages are preprocessed using the OAEP3 injective mapping before being encrypted with regular ElGamal: a ciphertext then looks like a pair $(g^r, OAEP3(m) \cdot h^r)$.

The proposed attack works by allowing any mixer, or indeed any party, to calculate the ciphertext of the mark of a previous mix using the homomorphic property of ElGamal. They are then free to use this ciphertext containing the previous mixer's mark to emulate the mixer and hence bypass these mixers without detection.

The initial input to the mix is a vector of N ciphertexts $c_1, \ldots, c_N$, containing the message $m_1, ..., m_N$ encrypted using randomness $r_1, ..., r_N$. If we multiply the ciphertexts together, we obtain $c_0^* = \prod_{i=1}^{N} c_i$, an encryption of the message $\prod_{i=1}^{N} OAEP3(m_i)$ using the randomness $\sum_{i=1}^{N} r_i$.

Consider the state after the first (presumed honest) mix, using a permutation π and whose mark was a_1, has occurred. We now have a vector of N ciphertexts $c'_1, ..., c'_N$, containing the message $a_1 \cdot OAEP3(m_{\pi_1}), ..., a_1 \cdot OAEP3(m_{\pi_N})$ using randomness $r'_1, ..., r'_N$. If we multiply the ciphertexts together, we obtain $c_1^* = \prod_{i=1}^{N} c'_i$, an encryption of the message $\prod_{i=1}^{N} a_1 \cdot OAEP3(m_i)$ using randomness $\sum_{i=1}^{N} r'_i$.

If we now take compute $c_{a_1^N} = c_1^*/c_0^*$ we have an encryption of the message a_1^N with randomness $\sum_{i=1}^{N} r_i - \sum_{i=1}^{N} r'_i$. Now, if the encryption is performed in a typical group of public prime order q, it is clear that $\gcd(N, q) = 1$, and it is therefore easy to compute $N^{-1} \bmod q$ (using the extended Euclidean algorithm) and to obtain an encryption of a_1 as $c_{a_1} = (c_{a_1^N})^{N^{-1}}$.

Now, c_{a_1} is precisely the rerandomization factor that an attacker would need to apply if he wants to bypass M_1. This attack continues to work for all following mixers. As a result, the last mixer M_k can de-anonymize all the ballots as follows: (1) Obtain encryptions $c_{a_1}, \ldots, c_{a_{k-1}}$ of all the marks produced by the previous mixers, as described above; (2) multiply all these ciphertexts, as well as fresh encryptions of its own mark, in order to obtain a rerandomization factor $c_a = Enc(pk, \prod_{i=1}^{k} a_i)$; (3) take all the ciphertexts that were the inputs of M_1, rerandomize them, multiply them by c_a, shuffle them, and output the resulting ciphertexts.

The resulting ciphertexts are perfectly valid and contain all the expected marks, but M_k knows the exact mapping between the ciphertexts submitted by the voters and those that will be decrypted by the trustees, since he is now the only person having actually shuffled those ciphertexts. Decryption would then break the secrecy of the votes for everyone.

4 Fixes

The attack described in the previous section relies upon the fact that product of all ciphertexts has a known relationship which allows computing an encryption of the mark. The intuition behind the fixes is to spoil this clean relationship. However, significant care must be taken not to introduce new problems while doing this.

4.1 Addition of dummy ciphertexts

A first possible approach is to require each mixer to add extra "decoy" ciphertexts to its output, placed in a random position, which would be an encryption of a specified plaintext (think "decoy mixer k") OAEP3-transformed, so that the ElGamal plaintext is unknown (the OAEP3 transformation is probabilistic). And adversary attempting to perform the attack described above would recover $Enc(a_1^N \cdot OAEP3(\text{"decoy mixer k"})$, from which—due to the probabilistic nature of OAEP3—it could not recover the mark.

Once mixing is over all the ciphertexts are decrypted. It is expected that the dummy ciphertexts initially fail to open. They are then isolated and checked to ensure that after removing the marks added by latter mix servers they decrypt to the expected message. If this check fails then an investigation is launched.

This fix prevents the attack described above, because the product of the output ciphertexts now contains the extra decoy ciphertext, whose content is unknown thanks to the probabilistic nature of OAEP3. As a result, the attacker becomes unable to obtain an encryption of the marks (raised to a known power): doing so would require guessing the position at which the dummy ciphertext is added. However, the probability of guessing this position wrongly is non-negligible, actually close to one for a large number of ciphertexts to be mixed. Specifically, if n is the number of senders and d is the number of decoys the adversaries chance is $1 - \binom{n+d}{d}$.

4.2 DDH in groups of unknown order

Another possibility to fix this issue is to prevent deriving an encryption of a_i from an encryption of a_i^N. This operation is easy to perform in the prime order groups that are typically used for ElGamal encryption.

The picture changes if we compute in groups of hidden order in which DDH is believed to be secure. A classical example [6, 4] of such groups would be the group of quadratic residues modulo an RSA modulus $n = pq$ such that $(p-1)/2$

and $(q-1)/2$ are prime. Here, the order of the DDH group is $(p-1)(q-1)/4$, which is unknown to anyone ignoring the factors of n, and extracting N-th roots becomes a hard problem.

This solution makes it possible to use the marked-mixnet protocol without any change, but requires the generation of an RSA modulus of unknown factorisation, which may be an inconvenience, even though it can be performed using standard MPC protocols.

5 Remarks on Security

In this short paper, we refer to the original Marked Mix-net paper for further security discussions: they remain valid for our modified version of the protocol, both when generalising to RCCA Encryption systems, and when we include the fixes to the attacks.

To achieve a quantum-safe system we need to use a quantum safe RCCA encryption system which meets our additional constraints. The only quantum safe RCCA encryption scheme [2] in the literature does not appear to work. It is an interesting area of future work to modify or create a PQ RCCA encryption scheme to meet our constraints. Since the mix-net construction is simple compared to full verifiable mix-nets, especially not employing non-interactive zero-knowledge proofs, the marked mix-nets provide a good opportunity for an efficient way of quantum-safe privacy-preserving mixing, and the first fix suggested above should work with any PQ safe encryption system.

A full detailed security analysis of marked mix-nets is out of the scope of this short paper and remains an important piece of future work.

6 Conclusion and Future Work

We have shown that marked mix-nets as original presented are not secure but that there are straightforward fixes for the construction to prevent this attack. We also show that it is straightforward to generalise the construction to work with most rerandomisable CCA cryptosystems. In particular, this will also enable quantum-safe versions of marked mix-nets. It is an area of ongoing work to rigorously define and prove the security of the marked mix-nets construction.

Acknowledgements This work was supported by the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint project SURCVS, and by Belgium Fonds de la Recherche Scientifique for the joint FNR/F.R.S.- FNRS project SeVoTe.

References

1. Aumann, Y., Lindell, Y.: Security against covert adversaries: Efficient protocols for realistic adversaries. In: Vadhan, S.P. (ed.) 4th Theory of Cryptography Conference, TCC. Lecture Notes in Computer Science, vol. 4392, pp. 137–156. Springer (2007)

2. Bansarkhani, R.E., Dagdelen, Ö., Buchmann, J.A.: Augmented learning with errors: The untapped potential of the error term. In: Financial Cryptography. Lecture Notes in Computer Science, vol. 8975, pp. 333–352. Springer (2015)
3. Benaloh, J., Byrne, M.D., Eakin, B., Kortum, P.T., McBurnett, N., Pereira, O., Stark, P.B., Wallach, D.S., Fisher, G., Montoya, J., Parker, M., Winn, M.: Star-vote: A secure, transparent, auditable, and reliable voting system. In: EVT/WOTE. USENIX Association (2013)
4. Boneh, D.: The decision diffie-hellman problem. In: Algorithmic Number Theory. Lecture Notes in Computer Science, vol. 1423, pp. 48–63. Springer (1998)
5. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: CRYPTO. Lecture Notes in Computer Science, vol. 2729, pp. 565–582. Springer (2003)
6. Damgård, I., Jurik, M.: A length-flexible threshold cryptosystem with applications. In: ACISP. Lecture Notes in Computer Science, vol. 2727, pp. 350–364. Springer (2003)
7. Faonio, A., Fiore, D.: Optimistic mixing, revisited. Cryptology ePrint Archive, Report 2018/864 (2018), https://eprint.iacr.org/2018/864
8. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In: TCC. Lecture Notes in Computer Science, vol. 2951, pp. 152–170. Springer (2004)
9. Pereira, O., Rivest, R.L.: Marked mix-nets. In: Financial Cryptography Workshops. Lecture Notes in Computer Science, vol. 10323, pp. 353–369. Springer (2017)
10. Phan, D.H., Pointcheval, D.: OAEP 3-round: A generic and secure asymmetric encryption padding. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 3329, pp. 63–77. Springer (2004)
11. Ryan, P.Y.A., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: Financial Cryptography Workshops. Lecture Notes in Computer Science, vol. 9604, pp. 176–192. Springer (2016)