Attacks on Karlsson and Mitrokotsa's Grouping-Proof-Distance-Bounding Protocol

Or why one should not trust crypto papers published in IEEE Communication Letters*

Roel Peeters, Jens Hermans and Aysajan Abidin

KU Leuven, ESAT/COSIC & Imec, Belgium
Firstname.Lastname@esat.kuleuven.be

Abstract. In the recent IEEE communication letter "Grouping-Proof-Distance-Bounding Protocols: Keep All Your Friends Close" by Karlsson and Mitrokotsa, a protocol for grouping-proof distance-bounding (GPDB) is proposed. In this letter, we show that the proof that is generated by the proposed GBDP protocol does not actually prove anything. Furthermore, we provide a construction towards a distance-bounding grouping-proof, however it remains unclear if one can ever truly combine (privacy-preserving) distance-bounding and a grouping-proof. As a bonus, we also discuss IEEE communication letters' poor standards regarding scientific ethics and responsibility.

1 Introduction

The concept of a grouping-proof (or yoking-proof) was introduced by Juels in [1] to prove that a pair of RFID tags has been scanned simultaneously. Directly from Juels paper: "Our particular aim is to permit tags to generate a **proof that is verifiable offline by a trusted entity, even when readers are potentially untrusted**." The concept of grouping-proofs was later generalised to also cover multiple tags. A formal model and some impossibility results for grouping-proofs were introduced by Hermans and Peeters [2] in 2012.

Very recently, Karlsson and Mitrokotsa proposed a grouping-proof distance-bounding (GPDB) protocol in [3], which should generate a verifiable proof that multiple tags were present simultaneously and within a certain distance from the reader. Their construction extends a privacy-preserving distance-bounding (DB) protocol previously proposed by Hermans *et al.* [4].

In this letter we show that the proposed GPDB protocol should never be used as the generated proof does not actually prove anything. Additionally, we provide some intuition why such a construction, combining (privacy-preserving) distance-bounding and a grouping-proof, cannot be built.

* see acknowledgements

2 The GPDB Protocol

The GPDB protocol proposed in [3] is shown in Fig. 1 and 2. We keep the notations used in the original paper (Table 1).

Table 1. Notations

Notation	Description
$\mathcal{P}$	Prover
$\mathcal{V}$	Verifier
$\mathbb{G}_\ell$	Subgroup of points on the elliptic curve of prime order ℓ
P	Generator point of $\mathbb{G}_\ell$
$\mathbb{Z}_\ell^*$	A multiplicative group of prime order ℓ
$\mathsf{xcoord}(\cdot)$	The x-coordinate of a point mod ℓ
T_s	A timestamp
$[x]_n$	n-bit representation of x
$\mathsf{Enc}[\cdot]$	Public key encryption

All provers $\mathcal{P}_j$ (for $j = 1 \dots m$) have a private key $x_j \in \mathbb{Z}_\ell^*$ and corresponding public key $X_j = x_j P \in \mathbb{G}_\ell$. The public keys of the provers are stored in the verifier $\mathcal{V}$'s database. The verifier $\mathcal{V}$ also has a private key $y \in \mathbb{Z}_\ell^*$ and corresponding public key $Y = yP \in \mathbb{G}_\ell$, which is known to all provers.

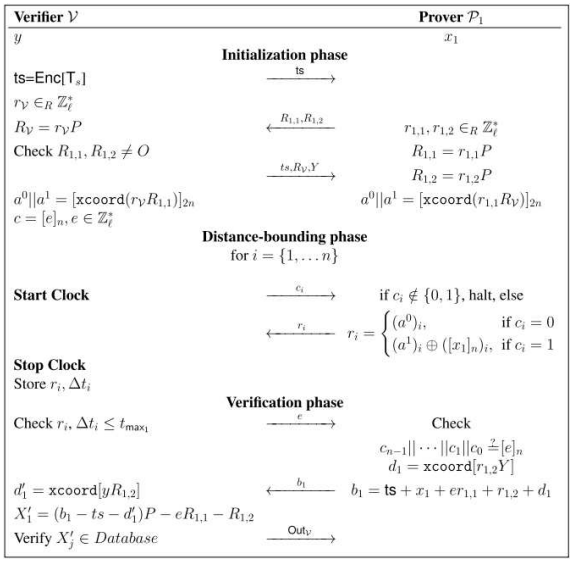

Fig. 1. The GPDB protocol for prover P_1 [3].

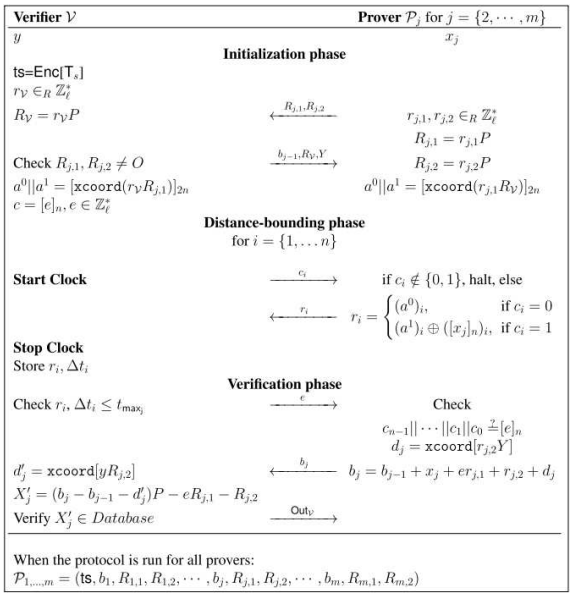

Fig. 2. The GPDB protocol for the rest of the provers [3].

In the first stage of the protocol (Fig. 1), $\mathcal{V}$ and the first prover execute the private distance-bounding protocol of [4] with one additional input, namely, an encrypted timestamp (unclear as to which key is used). In the second stage (Fig. 2), $\mathcal{V}$ repeats this protocol with each of the remaining provers but now the additional input is the output of the previous prover. Finally the proof is constructed as $\mathcal{P}_{1\ldots m} = (\mathsf{Ts}, b_1, R_{1,1}, R_{1,2}, \cdots, b_m, R_{m,1}, R_{m,2})$.

3 Flaws

Before going into the flaws of the generated proof by GPDP, let us take a closer look at the privacy-preserving distance-bounding protocol [4] that is used as a building block for GPDP. This distance-bounding protocol is a zero-knowledge interactive authentication protocol between a tag (prover) and the reader (verifier). At the end of the protocol run the reader will be convinced that the tag is within a certain distance. However, the transcript of this protocol run cannot be used to convince an offline party (party not present during the protocol run) that a given tag was within a certain distance of the reader or even present at the time when the protocol run took place. Note that this is not because of flaws in [4], but rather in the way it is used as a building block for GPDP.

The reason that the transcript cannot be used to convince an offline verifier that a tag was near the reader, is that an upper bound on the distance between reader and tag comes from measuring the time between sending and receiving

the fast bit exchanges (distance-bounding phase), something which is absent from the transcript.

The reason that the transcript cannot be used to convince an offline verifier that a tag was present, is that the used distance-bounding protocol is a zero-knowledge interactive authentication protocol, where the verifier only learns that prover is who it claims to be, but this cannot be convincingly shown to offline parties as the transcript can be constructed based only on public knowledge about the prover (i.e., it is not a signature). To illustrate the last point, we will show this for the classic Schnorr authentication protocol that consists of three phases:

1. Commit ($\mathcal{V} \leftarrow \mathcal{P}$): $R = rP$ with $r \in_R \mathbb{Z}_\ell^*$;
2. Challenge ($\mathcal{V} \rightarrow \mathcal{P}$): $e \in_R \mathbb{Z}_\ell^*$;
3. Response ($\mathcal{V} \leftarrow \mathcal{P}$): $b = ex + r$.

The verifier will verify that $R = bP - eX$ and hence be convinced that the prover has access to the private key x that corresponds to the public key $X = xP$. However by only knowing X, one can construct a protocol transcript (R, e, b), for which the above verification holds. This is done as follows: for any b and e, compute $R = bP - eX$. Similarly, for the privacy-preserving distance-bounding protocol [4], one can construct a transcript that withstands verification[1] as follows: for any b, e and r_2, compute $R_2 = r_2 P$ and $R_1 = e^{-1}\big((b - \text{xcoord}[r_j Y])P - X - R_2\big)$. Note that for constructing this transcript, only the public keys X and Y of the prover and verifier respectively are needed.

Now we will go into details of how the generated proof, which is basically the transcript of the protocol, by the GPDB protocol is flawed:

1. The proof cannot be verified Some crucial information to verify the X_j's is missing, namely the e_j's: $X_j = (b_j - b_{j-1} - d_j)P - \mathbf{e}Rj, 1 - R_{j,2}$. Note that e needs to be different for every distance-bounding protocol as the reader's challenges c_i are derived from e.

2. Verification requires the private key of the reader In order for an offline party to verify the X_j's in the given proof, it needs to compute $d_j = \text{xcoord}[yR_{j,2}]$, where y is the private key of the reader. Note that this private key cannot be the private key of the offline verifier, as this private key is needed by the reader during the protocol runs. As shown in paragraph 3, the reader is the only party that can verify the distance bound with the tags.

3. It is not an authentication proof As for the underlying privacy-preserving distance-bounding protocol [4], each run with a tag is a run of a zero-knowledge interactive authentication protocol. One can construct a valid transcript for a target tag with public key X as follows: for any b_{j-1}, b_j, e and $r_{j,2}$, compute

[1] $X = (b - d)P - eR1 - R_2$ with $d = \text{xcoord}[yR_2]$.

$R_{j,2} = r_{j,2}P$ and $R_{j,1} = e^{-1}\big((b_j - b_{j-1} - \text{xcoord}[r_{j,2}Y])P - X - R_{j,2}\big)$. This means that the transcript of the protocol run cannot be used to convince anyone (including the reader) that the tag (knowing the tag's private key x) was present.

4. It does not prove proximity at the time of the protocol run As for the underlying privacy-preserving distance-bounding protocol [4], timing information of the fast bit exchange rounds is not present in the transcript. Hence, from the generated proof, one cannot deduce an upper bound on the distance between the reader and tag. As a side note, given that each distance-bounding protocol is run separately and independently[2], one cannot simply assume that all tags are close to each other, but simply state that each tag $\mathcal{P}_j$ was close-by to the reader when its protocol was run.

Note that there is little use in adding timing information to the proof, as it would require trust in the reader that these timings are the ones measured during the protocol run (and not adjusted afterwards to come up with a more favourable distance bound). If one is to trust the reader, the reader could just as well state that a certain tag was within a certain distance bound, making the proof generated by the GPDP more compact and at the same time relieving the offline verifier from doing unnecessary computations.

5. It does not prove grouping From the generated proof, there is simply no way of knowing how much time there was in between each protocol run with the individual tags. With the chaining of protocol responses, using the output of the prior run as input for the current protocol run, these seem to be tied together. However a chain is susceptible to trimming (removing element from the end) and extension (adding new elements at the end) attacks. Given that tags can be removed and added again, the generated proof can be diverted in every possible way. Furthermore, due to the way this chaining is done for the GPDB protocol, one can even exchange elements anywhere in the chain without altering the rest of the chain.

Tags can be removed One can simply remove the last tag by removing $b_m, R_{m,1}, R_{m,2}$ from the proof.

Tags can be added One can simply add tags at the end by doing a legitimate protocol run with the target tag, which is given the last b_m as input. Furthermore, as shown in paragraph 3, one can also construct a valid transcript given the public key of the target tag.

Tags can be replaced Note that by applying the trick in paragraph 3, one can also simply replace a given tag anywhere in the chain, by keeping the b_j and b_{j-1} (and e_j) identical while only replacing $R_{j,1}$ and Rj_2.

[2] Adding a publicly known encrypted timestamp **ts** or prior response b_{j-1} to the response b_j does not change this, as these values are not directly coupled to the tag that is authenticated (see paragraph 3).

On the bright side it should be noted that privacy of the tags is fully preserved, unless someone wants to verify the proof and actually needs the reader's private key (the privacy model [5], in which privacy of the used distance-bounding protocol [4], assumes that the attacker has no access to the reader's private key). However even then, given how easy it is to divert the proof, it might be argued that privacy of the tags is preserved as there is no way of knowing if these were really present or not.

4 Listing Risks is Risky

Despite the number of critical flaws in the protocol, the GPDB paper contains a lengthy security analysis that conceals all of these. Looking deeper, it is clear that the paper never defines the security or privacy properties and only provides an informal listing of risks (Sect. II.C of [3]). Listing risks implies only listing a few potential attack strategies, which quite possibly do not cover the required security properties. This not only makes it impossible to understand what properties the protocol is supposed to achieve, it also rules out a formal assessment through proofs based on reductions.

The security analysis (Sect. IV of [3]) iterates over all the listed risks. The more lengthy analysis (proof) for distance fraud, mafia fraud, anonimity and traceability are simply repetitions of the proofs already provided for the underlying distance-bounding protocol [4] and thus not achievements of the newly proposed protocol. These properties still hold for the new protocol itself, however these do not extend to the generated proof, as shown in Sect 3.

In the end most of the flaws come down to a lack of formalism in definitions and an incorrect usage of the term grouping-proof. The protocol only provides distance-bounding and authentication of the tag if the reader is assumed to be trusted, which effectively downgrades the protocol to a plain privacy preserving distance-bounding protocol, just as the original work [4].

5 Towards building a distance-bounding grouping-proof

Hermans and Peeters [2] proposed impossibility results and provable constructions for grouping-proofs. A distinction is made between timed grouping-proofs and non-timed grouping-proofs. For a timed grouping-proof, one needs the cooperation of an actively participating trusted timestamping authority. For the non-timed proof, one can only prove that these tags were together at some point in time if all tags actively participate in each round of the protocol.

A distance-bounding grouping-proof should generate a proof showing that a number of tags are within a given small distance from each other and simultaneously scanned by a reader. The reader that acts as a verifier in the individual interactions with the tags (provers), now becomes the prover towards the offline party that verifies the generated proof. Implicitly this offline party will not trust the reader (otherwise there is no need for a proof) and only trust a subset[3] of

[3] If none of the participants are trusted, obviously the proof has no value.

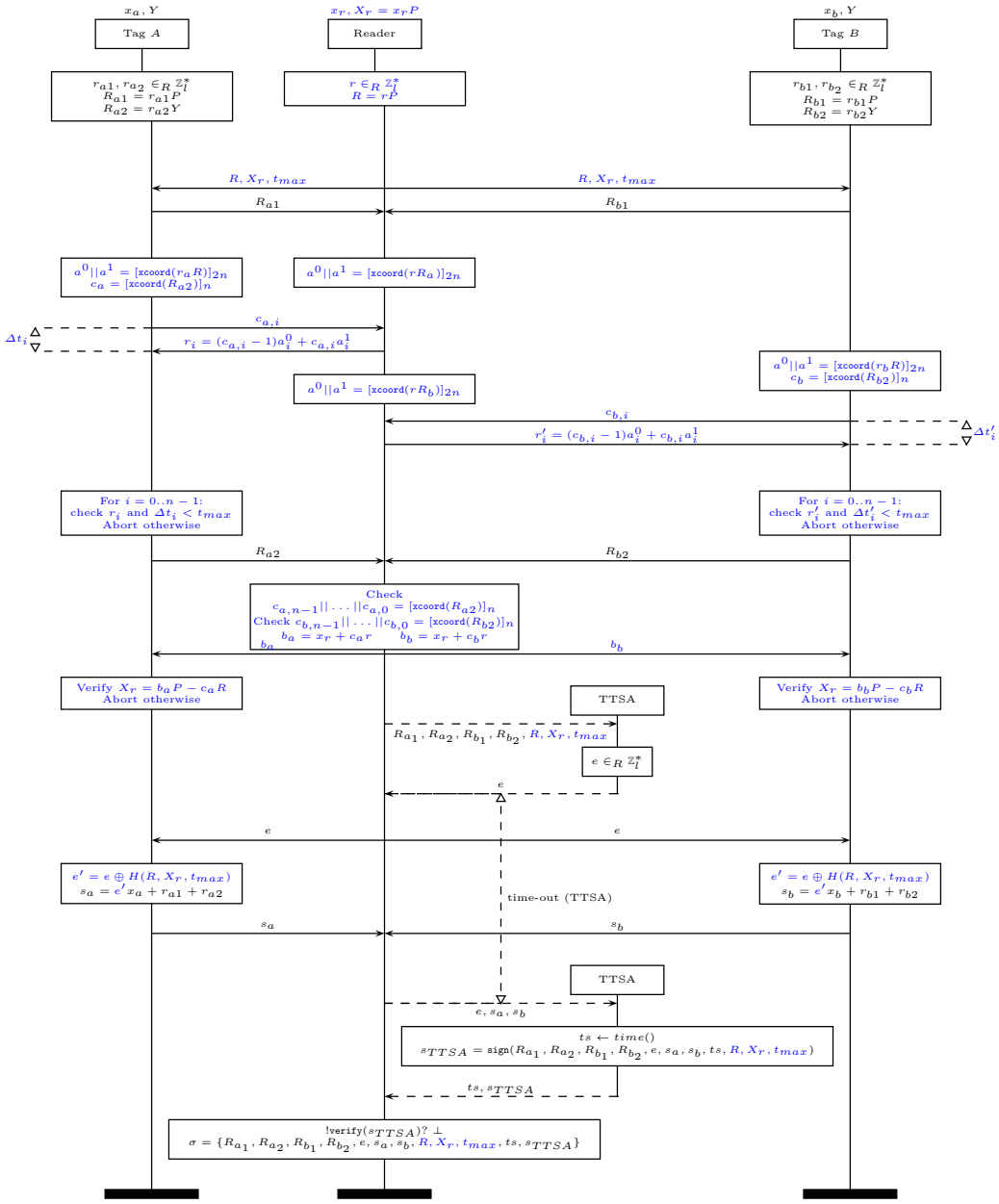

Fig. 3. Two-party grouping-proof protocol with timestamp [2], augmented with distance-bounding.

the tags. The idea being that an honest tag cannot be included in a malicious proof.

The above trust model means that if one is to include distance-bounding into a grouping-proof, it will need to be the reader that takes up the role of prover in the distance-bounding subprotocol. Now, the tag verifies that the reader is within a certain distance bound, and abort if not (ensuring that no false distance-bounding grouping-proof with an honest tag in it is generated). Since for the distance-bounding, it is now the reader that proves its position to the tag, privacy of the prover is no longer an issue as the reader is generally considered being public. Hence any secure (with respect to distance fraud and maffia fraud) distance-bounding protocol can be used as subprotocol. When plugging in a distance-bounding protocol in the proveably secure constructions of [2], one needs to ensure that the (possibly private towards a given off-line verifier) authentication of the tags is coupled to the distance-bounding protocol with the reader (including the upper distance bound or t_{max}).

Fig. 3 represents our proposed construction for two tags (however it can be trivially extended to m tags), making use of the construction with a trusted timestamp authority (TTSA) of [2], achieving private tag authentication, combined with a simplied version of [4] as the reader requires no privacy. The generated proof $\sigma = \{\mathbf{R_{a_1}}, \mathbf{R_{a_2}}, \mathbf{R_{b_1}}, \mathbf{R_{b_2}}, \mathbf{e}, \mathbf{s_a}, \mathbf{s_b}, \mathbf{R}, \mathbf{X_r}, \mathbf{t_{max}}, \mathbf{ts}, \mathbf{s_{TTSA}}\}$ is verified by a trusted offline party (with private key y) as follows: first check the signature s_{TTSA} of the TTSA, then verify the authentiction of the individual tags: $X_i = e'^{-1}(s_i P - R_{i_1} - y^{-1} R_{i_2})$ for $i \in \{a, b\}$, where $e' = e \oplus H(R, X_r, t_{max})$.

The proposed construction generates a proof that can convince an offline party that all tags were together at some (if a trusted time authority was actively involved, specific) point in time and that a subset of these tags were within a certain distance of the reader, even if this reader is not trusted. Note that this does not say anything about the distances between the tags, the reader could consist of several connected readers, each being physically close to honest tags for doing the time-critical rounds in the distance-bounding protocol.

6 Conclusion

"Grouping-Proof-Distance-Bounding Protocols: Keep All Your Friends Close" by Karlsson and Mitrokotsa should never be used since the generated proof does not prove anything: from this proof, one cannot be convinced that the tags were near the reader or even present at all at the time of running the GPDB protocol. Furthermore, it remains unclear if one can truly combine grouping-proofs with distance-bounding.

Acknowledgements

First of all we want to thank anonymous reviewers 3 and 4 for their efforts, the following rant does not apply to them. The simple truth is that [3] should not have been published to begin with, but given that it did, the journal should also

be able to accept criticism on it. IEEE communication letters failed to act in a responsible and/or ethical way by not accepting criticism on a paper which is clearly flawed, because the critism "uses harsh language" and because "the authors of [3] do not explictly state the aims of a grouping-proof generation protocol" (and hence should not be critized on this). The fact that the authors of [3] do not even clearly state the aims of such a grouping-proof generation protocol is the entire problem to begin with. Consider the following analogy:

1. Researchers A invent a new super elastic material and publish their research.
2. Other researchers B write a paper on how this new material could be used to build an unbreakable bridge (as it is super elastic). In their paper, they even mention the result of the original paper on the super elastic material showing that it remains super elastic even at -20 and +50 degrees Celcius.
3. The paper gets pulished by a journal.
4. Researchers A write a reaction paper and submit it to the same journal, stating that making a super elastic bridge is insane, as a bridge should also be sturdy to get people and goods safely accross the bridge and protect whatever is underneath the bridge from being crushed when some weight is put on the bridge.
5. The journal rejects the reaction paper as the original bridge paper did not state that their bridge was supposed to be sturdy and/or meant to transport good or people accross.
6. After a couple of years, someone decides to build the bridge as there seems to be no scientific objection to the original paper ...

Below, the direct quotation of the associate editor to notify us of the rejection.

> Your paper may not be resubmitted for review. The reasons for this are as follows: Based on the clear request in the last decision letter, many reviewers pointed out that the authors' answers were not convincing towards the presence of major flaws in the original protocol and proposing alternative solutions with enough contributions in the current letter, and I agree with them. The authors didn't demonstrate that the reader/verifier should be considered "Untrusted" in grouping-proofs. The quotation from [1] in the introduction states the particular aim of Juels' original paper on grouping-proofs, and does not appear to state a universal definition. As such, there is no strong ground to consider the assumption in the original paper [3] as wrong. Perhaps a weak assumption would be a better description. Reviewers 1 and 5 pointed out to new requirements added by the current letter authors, which the original protocol didnt claim to offer, hence cannot be considered as vulnerability or attack. Moreover, the current letter is not providing solution for these requirements. Few reviewers pointed out that the current letter still lacks sufficient contribution to justify a publication, and I agree with them.

After writing the following appeal to the journal's editor-in-chief:

> We believe that it is essential for the advance of science and for the scientific integrity and quality of your journal that it accepts letters that point out serious weaknesses in previously published letters.

In the area of cryptography, technical flaws or new weaknesses are discovered in existing schemes on a regular basis; this is not a flaw of the science, but crucial for scientific progress. Weaknesses are found even in the work of leading scientists in the field. In the cryptographic community, it is essential to publish cryptanalysis or "attack" papers rather than revoking the original paper, such that the community can learn from such shortcomings. Not publishing this letter will not make the flaws in the original letter go away. Nevertheless, the original letter will still enjoy some credit by being published in your journal. We wrote this reaction letter as we are clearly concerned that some of your readers who are less knowledgeable in the field of cryptography might be convinced that the original letter is a solid contribution to the field and hence might even consider using this kind of proof in real life scenarios.

In cryptographic protocols small subtleties can destroy essential security properties. Because of these subtleties, it is not always clear whether or not a protocol achieves its goals (as also witnessed by some of the reviewers' comments).

Next, we will explain the technical matter. The original letter by Karlsson and Mitrokotsa [3] constructs a grouping-proof based on the transcripts of the privacy-preserving distance-bounding protocol [4] (by Hermans, Peeters and Onete; two of which are the authors of this response letter!). Our letter is not about the underlying zero-knowledge privacy-preserving distance bounding protocol [4], which is fine. However, the transcript of such a protocol cannot be used as a proof since the protocol is zero-knowledge.

Our attack is thus aimed at the proof that is generated by running the protocol from [3]. Some reviewers seem to be only concerned about the protocol, and only take adversarial models during the protocol into account, while completely ignoring the fact that the purpose of the protocol in [3] is to generate a proof (for which another model applies). Even though [3] does not really go into the details of which party should verify the proof and how, we feel that it is only reasonable to assume that if one generates a proof, it is meant to be verified at some later point in time by a third party. We show in our letter that no one (not even the party generating the proof) can verify this proof at a later point in time.

Imagine the following scenario (from the original paper on grouping proofs of Juels [1]): Suppose that a prescription and the corresponding medicine must be scanned together at the pharmacy. The reader scans them both simultaneously and outputs a proof that these were together and close to each other at the time of distribution (the goal of the [3]). If the proof is not intended to be verified by a third party but only by the reader itself at the time of the protocol run, then the reader can just as well output a statement saying that the prescription and the corresponding medicine were scanned together. Nevertheless, as explained in Section III-3, no party, including the reader itself, can convince itself at a later stage that a certain tag (i.e. the prescription or medicine) was present, due to the zero-knowledge nature of the underlying distance-bounding proof.

We provided a construction that goes into the direction of what a distance-bounding grouping-proof generation protocol should look like, as this was requested by the reviewers. However, given the inherit properties of both types of constructions: grouping-proofs and distance-bounding protocols, this is not possible (as we also explain and justify in the letter).

> Based on the presented arguments, we would like to request you to revise the decision on whether or not to publish this letter or refer this letter to another editor.

we got back this message:

> Most of the appeal (and most of the content of the rejected paper) is based on the contention that the contribution of the original paper [3] should have been to generate a proof which is verifiable offline by a third party. However, [3] does not explicitly claim to do this, and it is reasonably clear from the exposition in [3] that the protocol therein does not seek to achieve this (at least 3 of the 5 reviewers, as well as the Associate Editor, mentioned that based on [3]). A new paper is then not needed to explain this fact. A paper which aims to improve upon the work of [3] should begin by explaining very briefly (and without harsh language) that [3] does not generate a proof that can be provided to a third party for verification. Then, the majority of such a paper should be devoted to new contributions. In the submitted work (CL2016-3013), most of the paper was concerned with pointing out (rather harshly, also in the opinion of most reviewers) what [3] did not achieve; this being said, the contribution beyond the work of [3] was considered by the reviewers and Associate Editor not to be sufficient for publication in IEEE Communications Letters.

References

1. A. Juels, "Yoking-proofs for RFID tags," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pp. 138–143.
2. J. Hermans and R. Peeters, "Private yoking proofs: attacks, models and new provable constructions," in *RFIDSec*, ser. LNCS. Springer, 2012, pp. 96–108.
3. C. Karlsson and A. Mitrokotsa, "Grouping-proof-distance-bounding protocols: Keep all your friends close," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1365–1368, 2016.
4. J. Hermans, R. Peeters, and C. Onete, "Efficient, secure, private distance bounding without key updates," in *ACM WiSec*. ACM, 2013, pp. 207–218.
5. J. Hermans, R. Peeters, and B. Preneel, "Proper RFID Privacy: Model and Protocols," *IEEE Transaction on Mobile Computing*, vol. 13, no. 12, pp. 2888–2902, 2014.