Breaking and Fixing the Security Proof of Garbled Bloom Filters

Cédric Van Rompay, Melek Önen

Breaking and Fixing the Security Proof of Garbled Bloom Filters

Cédric Van Rompay[1] ✉ and Melek Önen[1]

EURECOM, France
`{vanrompa,onen}@eurecom.fr`

Abstract. We identify a flaw in the proof of security of Garbled Bloom Filters, a recent hash structure introduced by Dong et al. (ACM CCS 2013) that is used to design Private Set Intersection (PSI) protocols, a important family of protocols for secure cloud computing. We give counter-examples invalidating a claim that is central to the original proof and we show that variants of the GBF construction have the same issue in their security analysis. We then give a new proof of security that shows that Garbled Bloom Filters are secure nonetheless.

Keywords: garbled bloom filter · private set intersection · provable security

1 Introduction

Private Set Intersection (PSI) protocols is one of the most important family of protocol for secure computation that plays a central role in cloud computing (see Section 1 of [4]). Garbled Bloom Filters (GBF) are a recent hash structure introduced by Dong et al. in [4] (ACM CCS 2013) that are useful in the design of PSI protocols. The idea of GBF is to combine a Bloom Filter (BF) with XOR-based secret sharing to enable efficient test membership with regard to a set while hiding the presence of elements in this set that were not searched for. The construction of Dong et al. had quite a large impact in research as it was used [3,15–17], improved [11,12] and cited [5,7,9,10,14] numerous times already.

The proof of Dong et al. can be summarized the following way: In a first part they use a property of Bloom Filters to show that some event happens with negligible probability; then in a second part they assume the absence of the previously mentioned event and invoke the security of XOR-based secret sharing to conclude the proof. This invocation of the security of XOR-based secret sharing is done in a very immediate way, neglecting the fact that the functioning of GBF, however heavily inspired by the XOR-based secret sharing scheme, is not strictly speaking an instance of this scheme. The same remarks hold for a PSI protocol suggested by Pinkas et al. in [11], based on the original GBF construction by Dong et al., for which the proof of security is very short and follows a reasoning similar to the one of Dong et al.

In this paper we show that a simple invocation of the security of XOR-based secret sharing is in fact not sufficient to show that GBFs are secure. We do so

by providing a counter-example, and further a larger class of counter-examples, that invalidate the claims made in both previously mentioned proofs.

We show however that GBFs do satisfy their claimed security properties by providing a new, more rigorous proof.

1.1   Organization of the Paper

In Section 2 we describe Bloom Filters, Garbled Bloom Filters as introduced by Dong et al. in [4], and the original proof of Dong et al. for the security of Garbled Bloom Filters. In Section 3 we give a counter example (and a class of counter-examples) that invalidates the proof of Dong et al. In Section 4 we describe the impact of our results on other GBF constructions that were inspired by the one of Dong et al. In Section 5 we give a new proof of security for the GBF construction of Dong et al. Finally in Section 6 we compare this work with related work.

2   Preliminaries

2.1   Notations

We make use of the following usual conventions: With X a set, we denote by $x \xleftarrow{\$} X$ the fact that x is sampled uniformly from X. With i a positive number, we denote by $[i]$ the sequence $(1, 2, \ldots, i)$. A function $\mu(\cdot)$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large n, it holds that $\mu(n) \leq 1/p(n)$. Throughout the paper, λ denotes the security parameter and two probability ensembles $\mathcal{X} = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are said to be *computationally indistinguishable* [6, Definition 7.30] denoted $\mathcal{X} \stackrel{c}{\equiv} \mathcal{Y}$, if for every probabilistic polynomial-time distinguisher D there exists a negligible function μ such that:

$$\left| Pr[D(1^\lambda, x) = 1; x \xleftarrow{\$} X_\lambda] - Pr[D(1^\lambda, y) = 1; y \xleftarrow{\$} Y_\lambda] \right| \leq \mu(\lambda)$$

2.2   Bloom Filters

Bloom Filters (BFs), introduced by Bloom in [1] and further studied by Broder and Mitzenmacher in [2], are a hash structure that aims at efficiently testing membership in a set. A BF is an array B of M bits associated with k random hash functions $h_1, \ldots, h_k : \{0, 1\}^* \to [M]$. B is initialized by setting all the array values to zero and one inserts an element $x \in S$ in B by setting $B[h_i(x)]$ to 1 for all i. Finally one checks the presences of x in the set S encoded by B by testing whether $B[h_i(x)]$ is equal to 1 for all $i \in [k]$; if it is not the case then x cannot be in S, otherwise x is in S with high probability. Following [12] we use the notation h_* to denote the set of all positions corresponding to an element x or a set S:

$$h_*(x) = \{h_i(x) \ \forall i \in [k]\}$$

$$h_*(S) = \bigcup_{x \in S} h_*(x)$$

We will denote by BF_S the Bloom Filter encoding set S when there is no ambiguity about what parameters M and $(h_i)_{i \in [k]}$ where used.

The event that x appears to be encoded in B while it is actually not in S is called a *false positive*. Dong et al. [4] show that the probability for $x \notin S$ to cause a false positive is negligible in the number of hash functions k. As a consequence, setting the number of hash function as greater or equal to the security parameter (which is what Dong et al. do) results in a false positive probability negligible in the security parameter.

Broder and Mitzenmacher in [2] show that the optimal value for k, that minimizes the false positive probability for a given M and set size N, is:

$$k = \ln(2) \frac{M}{N} \ . \tag{1}$$

They also show that with this value of k about half of the bits are set after insertion of all the elements in S.

2.3   Garbled Bloom Filters

Garbled Bloom Filters (GBFs) were introduced by Dong et al. in [4] (ACM CCS 2013). GBF is a variant of BF that has some security properties making it suitable for the design of Private Set Intersection (PSI) protocols (see [11] for a description of PSI and a review of most recent schemes, including the one of [4]).

Like a BF, a GBF is an array of length M associated with k random hash functions $h_1, \ldots, h_k : \{0,1\}^* \to [M]$. However the components of a GBF are not bits but bit strings of length λ. One inserts x in a GBF B by ensuring that $\bigoplus_i B[h_i(x)] = x$, and checks the presence of x in B by testing the same equality.

During insertion, each share is picked uniformly at random as in a XOR secret sharing scheme, except the shares that were already set by the insertion of a previous element that are left unchanged. Components that were never wrote during insertion of the whole set are filled with random values. Algorithm 1 gives a more formal description of how a GBF is built. As with "normal" Bloom Filters, we will denote by GBF_S a GBF encoding set S when there is no ambiguity as to the parameters used.

The security property of GBFs, which will be given formally in Section 2.5, can be informally described as follows:

Definition 1 (Security of GBF – informal). *Let S and C be two sets; Given only $\{GBF_S[i] \ \forall i \in h_*(C)\}$ one cannot get any information about $S - C$.*

2.4   Private Set Intersection Based on GBF

We give a quick overview of how GBFs are used in the design of Private Set Intersection (PSI) protocols with one-sided output. Informally, a PSI protocol

---

Algorithm 1: An algorithm for building a GBF representing set S with parameters $M, (h_i)_{i=1...k}, \lambda$

> **Algorithm:** GBF.Build
>
> **Input:** $S, M, (h_i)_{i=1...k}, \lambda$
> **Output:** B
> Initialize B as an empty array of length M ;
> **for** $x \in S$ **do**
> > **if** $\exists j \in h_*(x) \ : \ (B[j]$ *is empty*) **then**
> > > **for** $i \in h_*(x) - \{j\}$ **do**
> > > > **if** $B[i]$ *is empty* **then**
> > > > > $B[i] \xleftarrow{\$} \{0,1\}^\lambda$;
> > >
> > > set $B[j] \leftarrow x \oplus \bigoplus_{i \in h_*(x) - \{j\}} B[i]$;
> >
> > **else**
> > > Abort. ;
>
> Fill any remaining empty component with fresh random values ;

---

is a protocol between two parties, each having a set, who want to compute the intersection of their respective sets without revealing more information than this intersection. In the one-sided output setting only one party, called the *receiver*, learns the intersection, while the other party, called the *sender*, learns nothing.

In the PSI protocol of Dong et al. [4], the sender holds a set S and computes GBF_S while the receiver holds a set C and computes BF_C. Both parties use the same (G)BF parameters.

The two parties then run an Oblivious Transfer (OT) protocol, that is a protocol that allows a party (called *receiver* and that matches the receiver of the PSI protocol) to retrieve a record in a database held by another party (the *sender*, who again matches the sender of the PSI protocol) without revealing to the sender which record was retrieved by the receiver and without revealing to the receiver the other records in the database.

The OT protocol is used in the PSI protocol of [4] by the receiver in order to retrieve the components of GBF_S corresponding to the "ones" in BF_C. For any element in $S \cap C$, its corresponding components were retrieved so the receiver is able to assert its presence in $S \cap C$. At the same time, the security property of GBFs guarantee that the receiver got no information about any element of $S - C$. As for the sender, the privacy properties of the OT protocol suffice to prevent him from learning anything about the set C of the receiver.

2.5   Original Proof of Security by Dong et al. [4]

The security of GBF is expressed by Theorem 4 in [4] which we reformulate in an equivalent way in Theorem 1 of this paper. This theorem requires the definition of the intersection between a GBF and a BF sharing the same parameters (see Section 4.2 of [4])

Definition 2 (Intersection between a GBF and a BF). *Let* $M, (h_i)_{i=1...k}$ *and* λ *be some GBF parameters. Let* S *and* C *be two sets, and let* GBF_S *and* BF_C *be built with parameters* $M, (h_i)_{i=1...k}$ *(and* λ *for the GBF). The intersection of* GBF_S *and* BF_C, *noted* $GBF_S \cap BF_C$, *is defined as:*

$$(GBF_S \cap BF_C)[i] \leftarrow \begin{cases} GBF_S[i] & \text{if } BF_C[i] = 1 \\ a \text{ } random \text{ } value & otherwise \end{cases}$$

Dong et al show that $GBF_S \cap BF_C$ is a correct GBF encoding $S \cap C$. We also define the notion of "extraction" of a GBF with a BF, which is equivalent to the notion of intersection but will make our proof in Section 5 simpler. We will use the notion of intersection mostly in Section 3 in order to stay as close as possible to the notation of Dong et al., and in Section 5 we will mostly use the notion of extraction. With extraction, "non-selected" components are simply dropped, or equivalently set to a special "empty" value, instead of being replaced by a random value. It should be obvious that one obtains as much information from a uniform independent random value than from a fixed value.

Definition 3 (Extraction of a GBF with a BF). *Let* $M, (h_i)_{i=1...k}$ *and* λ *be some GBF parameters. Let* S *and* C *be two sets, and let* GBF_S *and* BF_C *be built with parameters* $M, (h_i)_{i=1...k}$ *(and* λ *for the GBF). The extraction of* GBF_S *using* BF_C, *noted* $\mathsf{Extract}(BF_C, GBF_S)$, *is defined as:*

$$\mathsf{Extract}(BF_C, GBF_S)[i] \leftarrow \begin{cases} GBF_S[i] & \text{if } BF_C[i] = 1 \\ empty & otherwise \end{cases}$$

We now give Theorem 4 of [4] in a slightly reformulated but equivalent form:

Theorem 1 (Security of GBF (Theorem 4 of [4])). *Let* λ *and* $N \in \mathbb{N}$ *and let* $k = \lambda$ *and* $M = Nk/\ln(2)$; *let* $(h_i)_{i \in [k]}$ *be a sequence of random oracles* $\{0,1\}^* \to [M]$. *we have*

$$(S, C, GBF_S \cap BF_C) \overset{c}{\equiv} (S, C, GBF_{S \cap C} \cap BF_C)$$

Where S *and* C *have at most* N *elements. Equivalently with our "extraction" notation:*

$$(S, C, \mathsf{Extract}(BF_C, GBF_S)) \overset{c}{\equiv} (S, C, \mathsf{Extract}(BF_C, GBF_{S \cap C}))$$

The proof Dong et al. give for Theorem 1 is reproduced below, with only minor modifications to make it match our notation. Namely, what is written $GBF_{C \cap S}$, $GBF'_{C \cap S}$ and $GBF''_{C \cap S}$ in the original text is written respectively $GBF_S \cap BF_C$, $GBF_{S \cap C}$ and $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ in ours. We give a quick overview of their proof: In their "case 1", they show that the probability that some element of $S - C$ has all its positions in $h_*(C)$ is negligible; then in "case 2" they argue that if no element of $S - C$ has all its elements in $h_*(C)$, the distribution of $GBF_S \cap BF_C$ is then identical to the one of $GBF_{S \cap C}$. They invoke *"the security*

of the XOR-based secret sharing scheme" to argue that an element of $S - C$ of which one of the shares was re-randomized during intersection cannot leave any trace in the resulting GBF (this is the argument we will go against in Section 3).

(Proof of Theorem 1 as it appears in [4])
Given $GBF_S \cap BF_C$, we modify it to get $(GBF_S \cap BF_C) \cap BF_{S \cap C}$. We scan $GBF_S \cap BF_C$ from the beginning to the end and for each location i, we modify $(GBF_S \cap BF_C)[i]$ using the following procedure:
- If $(GBF_S \cap BF_C)[i]$ is a share of an element in $C \cap S$, then do nothing.
- Else if $(GBF_S \cap BF_C)[i]$ is a random string, do nothing.
- Else if $(GBF_S \cap BF_C)[i]$ is a share of an element in $S - C \cap S$, replace it with a uniformly random λ-bit string.

The result is $(GBF_S \cap BF_C) \cap BF_{S \cap C}$. Every $(GBF_S \cap BF_C)[i]$ must fall into one of these three cases, so there is no unhandled case.
Now we argue that the distribution of $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ is identical to $GBF_{S \cap C}$. To see that, let's compare each location in $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ and $GBF_{S \cap C}$. From Algorithm 1 and the above procedure, we can see that $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ and $GBF_{S \cap C}$ contain only shares of elements in $S \cap C$ and random strings. Because $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ and $GBF_{S \cap C}$ use the same set of hash functions, for each $0 \leq i \leq m - 1$, $((GBF_S \cap BF_C) \cap BF_{S \cap C})[i]$ is a share of an element in $C \cap S$ iff $GBF_{S \cap C}$ is a random string. The distribution of a share depends only on the element and the random strings are uniformly distributed. So the distribution of every location in $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ and $GBF_{S \cap C}$ are identical therefore the distributions of $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ and $GBF_{S \cap C}$ are identical.
Then we argue that the distribution of $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ is identical to $GBF_S \cap BF_C$ except for a negligible probability η.
Case 1, $GBF_S \cap BF_C$ encodes at least one elements in $S - C \cap S$. In this case the distribution of $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ differs from the distribution of $GBF_S \cap BF_C$. From Theorem 3, the probability of each element in $S - C \cap S$ being encoded in $GBF_S \cap BF_C$ is ϵ. Since there are $d = |S| - |C \cap S|$ elements in $S - C \cap S$, the probability of at least one element is falsely contained in $GBF_S \cap BF_C$ is:

$$\eta = \text{[skipped...]} \leq 2d\epsilon$$

As we can see η is negligible if ϵ is negligible.
Case 2: $GBF_S \cap BF_C$ encodes only elements from $C \cap S$. In this case, each element of $S - C \cap S$ may leave up to $k - 1$ shares in $GBF_S \cap BF_C$. The only difference between $GBF_S \cap BF_C$ and $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ is that in $(GBF_S \cap BF_C) \cap BF_{S \cap C}$, all "residues" shares of elements in $S - C \cap S$ are replaced by random strings. From the security of the XOR-based secret sharing scheme, the residue shares should be uniformly random (otherwise they leak information about the elements). Thus the

procedure does not change the distribution when modifying $GBF_S \cap BF_C$ into $(GBF_S \cap BF_C) \cap BF_{S \cap C}$. So the distributions of $GBF_S \cap BF_C$ and $(GBF_S \cap BF_C) \cap BF_{S \cap C}$ are identical. The probability of this case is at least $1 - \eta$.

Since $(GBF_S \cap BF_C) \cap BF_{S \cap C} \equiv GBF_{S \cap C}$ always holds and $GBF_S \cap BF_C \equiv (GBF_S \cap BF_C) \cap BF_{S \cap C}$ in case 2, we can conclude that $Pr[GBF_S \cap BF_C \equiv GBF_{S \cap C}] \geq 1 - \eta$ thus

$|Pr[D(GBF_S \cap BF_C) = 1] - Pr[D(GBF_{S \cap C}) = 1]| \leq \eta.$ $\square$

3   Invalidation of the Proof in [4]

The end of the proof contains the following assertion: "$(GBF_S \cap BF_C) \cap BF_{S \cap C} \equiv GBF_{S \cap C}$ always holds and $GBF_S \cap BF_C \equiv (GBF_S \cap BF_C) \cap BF_{S \cap C}$ holds in case 2". This should result in $GBF_{S \cap C} \equiv GBF_S \cap BF_C$ in case 2. We invalidate this claim by giving a counter-example. Let the number of hash functions be $k = 3$; let x and y be two elements of $S - C$ such that $h_1(x) = h_1(y)$ and that for all $i \neq 1$, $h_i(x) \in h_*(C)$ and $h_i(y) \in h_*(C)$. This example is illustrated in Figure 1. Note that this example can be situated in the case 2 of the proof of [4] as it does not require any element of S to have all its positions in $h_*(C)$.
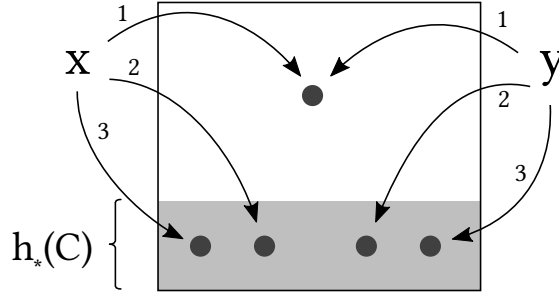
Fig. 1: Illustration of our counter-example.

We have that GBF_S must satisfy the following equations where we note $GBF_S[h_i(x)]$ as x_i (and similarly with y):

$$x_1 \oplus x_2 \oplus x_3 = x \qquad (2)$$
$$y_1 \oplus y_2 \oplus y_3 = y \qquad (3)$$
$$x_1 = y_1 \qquad (4)$$

Combining (2), (3) and (4) gives:

$$x \oplus y = x_1 \oplus x_2 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3$$
$$x \oplus y = x_2 \oplus x_3 \oplus y_2 \oplus y_3$$

If we re-write the latter equation without our short-hand notation, we have that GBF_S satisfies the following:

$$x \oplus y =$$
$$GBF_S[h_2(x)] \oplus GBF_S[h_3(x)] \oplus GBF_S[h_2(y)] \oplus GBF_S[h_3(y)] \qquad (5)$$

Regarding $GBF_S \cap BF_C$, it does not satisfy equations (2) and (3) anymore because the component $GBF_S[h_1(x)]$ was replaced by a fresh random value during the intersection operation; but it still satisfies equation (5) as it only involves components that were not re-randomized during intersection, thanks to the fact that $h_2(x), h_3(x), h_2(y)$ and $h_3(y)$ are in $h_*(C)$.

On the other hand $GBF_{S\cap C}$, which was built without the knowledge of x and y, does not satisfy (5) (except with a very small probability). As a result a GBF where relation (5) *does not* hold is a valid outcome for the distribution of $GBF_{S\cap C}$ but not for the distribution of $GBF_S \cap BF_C$. Those distributions cannot be identical, and the proof given in [4] of Theorem 1 is wrong. The same counter-example can also be used to invalidate the claims that "$GBF''_{C\cap S} \equiv GBF'_{C\cap S}$" and that "$GBF_{C\cap S} \equiv GBF''_{C\cap S}$".

This is not just a typo in [4], but truly a flaw in the proof. Recall, the proof uses the fact that any $x \in S - C$ has, with overwhelming probability, one of its positions, say $h_1(x)$, out of $h_*(C)$. As a result this component is overwritten during intersection (or never retrieved in a PSI scenario). Dong et al. then invoke *"the security of the XOR-based secret sharing scheme"* to argue that no information can be obtained about $x_1 \oplus x_2 \oplus x_3$. But the GBF construction is not the exact same thing as a XOR secret sharing scheme, and the argument does not hold. More precisely, in a GBF the component $GBF_S[h_1(x)]$ (or x_1) may not be independent from other components in the GBF and in particular its value can be tied to the value of other components that may be in $h_*(C)$ and are thus "visible", which is the case with components y_2 and y_3 in our example.

3.1   Generalization of the Counter-Example

We give a larger class of situations where the same claims prove wrong. Let $P(S,C)$ (or just P if there is no ambiguity about the inputs) be the set of positions that appear an odd number of times in $(h_*(x) \ \forall x \in S - C)$:

$$P(S,C) = \{p \in h_*(S-C) \ : \ |\{(x,i) \in (S-C) \times [k] \ : \ h_i(x) = p\}| \bmod 2 = 1\}$$

Then GBF_S satisfies the following relation, of which (5) is a special case, and which is obtained the same way as (5) was obtained:

$$\bigoplus_{S\cap C} x = \bigoplus_{p\in P} GBF_S[p] \qquad (6)$$

If moreover $P \subset h_*(C)$, none of the concerned components are re-randomized during intersection so $GBF_S \cap BF_C$ satisfies the same relation, that is:

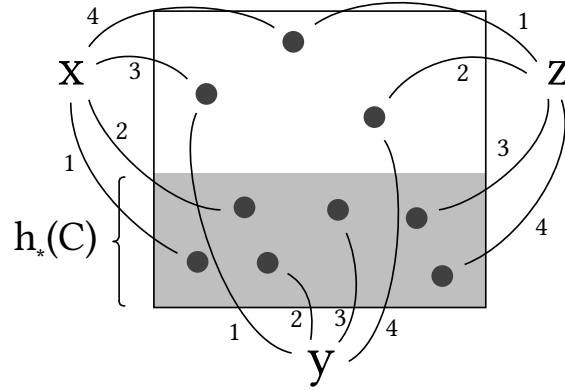$$\bigoplus_{S\cap C} x = \bigoplus_{p\in P} (GBF_S \cap BF_C)[p] \qquad (7)$$

Fig. 2: An example of a more general counter-example involving 3 elements of $S - C$.

Figure 2 illustrates such a more general case with 4 hash functions and involving 3 elements x, y and z where $GBF_S \cap BF_C$ would satisfy the following relation (but $GBF_{S \cap C}$ would not):

$$x \oplus y \oplus z = x_1 \oplus x_2 \oplus y_2 \oplus y_3 \oplus z_3 \oplus z_4$$

4   Other GBF Constructions

We describe the consequences of our findings on the other GBF constructions that were inspired by the one of Dong et al., namely the ones of Pinkas et al. [11, Section 4.3] (USENIX Security 2014), and Rindal and Rosulek [12] (EUROCRYPT 2017).

4.1   Pinkas et al. [11]: Same Situation as Dong et al. [4]

The construction of Pinkas et al. presents many optimizations over the one of Dong et al., for instance through the use of *random OT* instead of "classical" OT, but it also has a more essential difference with the construction of Dong et al. in that the sum of the components associated to an element need not be equal to the element itself. Instead, all component values are all chosen uniformly at random and the sender sends for each element in her set a "summary value" that is the sum of the components corresponding to this element, that is:

$$\{\bigoplus_i GBF_S[h_i(s)] \quad \forall s \in S\}$$

The receiver retrieves the components corresponding to her own elements via OT and compute similar sums for these elements. Finally, the receiver compares the sums she computed with the sums she received to learn which elements are in both sets.

The reasoning used by Pinkas et al. to show the security of this construction is similar to the reasoning of Dong et al., namely, that unless there is a $s \in S$ such that $h_*(s) \subset h_*(C)$ then the view of the receiver is not just computationally indistinguishable from a simulated view, but truly independent from $S - C$.

Unfortunately the same problem appears as with the proof of Dong et al. Take for instance the example of Figure 1: if x and y are in the sender's set S, then the receiver must have received these two values:

$$K_x = GBF_S[h_1(x)] \oplus GBF_S[h_2(x)] \oplus GBF_S[h_3(x)]$$
$$K_y = GBF_S[h_1(y)] \oplus GBF_S[h_2(y)] \oplus GBF_S[h_3(y)]$$

and because $h_1(x) = h_1(y)$ the XOR of these two received values is equal to:

$$
K_x \oplus K_y =
$$
$$
GBF_S[h_2(x)] \oplus GBF_S[h_3(x)] \oplus GBF_S[h_2(y)] \oplus GBF_S[h_3(y)]
\tag{8}
$$

Which only depends on values the receiver knows. The receiver is then able to detect the presence of x and y in $S - C$ by testing whether any two summary values have their sum equal to (8).

Again, and as with the GBF construction of Dong et al. [4], the construction of Pinkas et al. is actually secure since our new proof given in Section 5 should also apply. This means that such S and C are actually very hard to find. Nevertheless this shows that the security of the GBF construction of [11] cannot be proven simply by invoking the low probability to have $h_*(s) \subset h_*(C)$ for some $s \in S$.

4.2   Rindal and Rosulek [12]: no Apparent Issues

Rindal and Rosulek [12] present a new PSI protocol following the idea of the GBF-based PSI protocol of Pinkas et al. [11, Section 4.3] we presented in the previous section. They keep most of the ideas of Pinkas et al., including the use of random OT and the optimizations it enables, as well as the idea of having the sender sending summary values. However they build these summary values in a different way, which is essentially:

$$K_x = H\left(x \;||\; \bigoplus_{i \in h_*(x)} GBF_S[i] \right)$$

Where H is a secure hash function and $||$ denotes concatenation.

Interestingly, the presence of the hash function breaks the algebraic properties of the summary values that were used in the previous section, meaning that all the counter-examples we gave so far do not apply on the construction of Rindal and Rosulek. This construction may thus be secure even against someone knowing a subset $X \subset S - C$ such that $P(X, C) \subset h_*(C)$. But more importantly, the security proof Rindal and Rosulek give for their construction [12, Section 5.3] differs a lot from the proofs of Dong et al. [4] and of Pinkas et al. [11], and we

did not find in the paper of Rindal and Rosulek the issue we identified in [4] and [11].

Note however that the construction of Rindal and Rosulek and of Pinkas et al. cannot always be used as a drop-in replacement of the original construction of Dong et al. One example is a Searchable Encryption protocol [13] that uses Garbled Bloom Filters but where the receiver looks up several GBFs and must be unable to know what response (in the form of components retrieved) comes from what filter. This requires that the receiver must be able to decide on the result of a lookup ("present" or "absent") using only the components retrieved and without remembering what was the component that was being looked for. The authors modify the GBF construction of Dong et al. by having the components corresponding to an element having their sum equal to a fixed value instead of the value of the element itself:

$$\bigoplus_{i \in h_*(C)} GBF_S[i] = 0$$

Such a property could not be reached in a trivial way using the construction of Rindal and Rosulek (or even the one of Pinkas et al.) because the sending of summary values by the sender requires that the receiver knows what to compare these values with, which requires that the receiver knows what GBF the values correspond to. This shows why the study of the security proof of constructions other than the one of Rindal and Rosulek is still relevant.

5   New Proof of Security

5.1   New Case Distinction

Our proof follows the idea of the proof of Dong et al. [4]: we consider two cases, one that occurs with negligible probability and one in which the two distributions are actually identical, and this results in the two distributions being indistinguishable. What differs between our proof and the one of [4] is the case separation: as we saw, the assumption of case 2 of [4] that no element in $S - C$ has all its positions in $h_*(C)$ does not suffice to have $GBF_S \cap BF_C \equiv GBF_{S \cap C}$.

Instead, we make the following remark: it is very unlikely that there is some subset X of $S - C$ such that all the positions in $h_*(X)$ being mapped by a single element in X happens to be in $h_*(C)$. Said differently, for any subset $X \subset S - C$ there is at least one position in $h_*(X)$ that is both out of $h_*(C)$ and corresponds to a single element of X. Note that this covers the situation described in Section 3.1 (an thus our counter-examples in Figures 1 and 2 too): if all the position in $h_*(S - C)$ mapped an odd number of times are in $h_*(C)$, then all the positions out of $h_*(C)$ are mapped at least 2 times. Formally we define the *mapped-once positions of* X, noted $\boldsymbol{m}(X)$, and the *never-mapped positions of* X, noted $\boldsymbol{n}(X)$, as follows:

Definition 4 (Mapped-once and never-mapped positions of a set). *Let* $X \subset \{0,1\}^*$; *the set of mapped-once positions of X is defined as:*

$$\boldsymbol{m}(X) := \{p \in h_*(X) \ : \ \exists!(x,i) \in X \times [k] \ : \ h_i(x) = p\}$$

Similarly, the set of never-mapped positions of X is defined as:

$$\boldsymbol{n}(X) := \{p \in h_*(X) \ : \ \nexists(x,i) \in X \times [k] \ : \ h_i(x) = p\}$$

The never-mapped positions of X correspond to the zeroes in BF_X.

We then have the following:

Theorem 2. *Let $X \subset S - C$, then:*

$$P[\boldsymbol{m}(X) \subset h_*(C)] \leq negl(\lambda) \tag{9}$$

Proof: We explain why Equation (9) holds. Our explanation is in two parts: First we argue that the size of $\boldsymbol{m}(X)$ must be of size larger than k; from then the probability that all these positions are in $h_*(C)$ is lesser than the probability for one element to have all its positions in $h_*(C)$, which is already negligible (proved by Dong et al. in their "case 1"). We start by showing that $|\boldsymbol{m}(X)| \geq k$ with overwhelming probability. Consider the sequence of sets $X_1, X_2, \ldots, X$ where each set has one more element than the previous one. The number of mapped-once positions of $X_i = X_{i-1} \cup \{x\}$ for some i and some x is then the number of mapped-once positions of X_{i-1} plus the positions of x that are in $\boldsymbol{n}(X_{i-1})$ (some new mapped-once positions), minus the positions of x that are in $\boldsymbol{m}(X_{i-1})$ (positions that are not mapped-once anymore). Statistically, we thus have the following expected difference:

$$E[|\boldsymbol{m}(X_i)| - |\boldsymbol{m}(X_{i-1})|] = k\frac{\boldsymbol{n}(X_{i-1})}{M} - k\frac{\boldsymbol{m}(X_{i-1})}{M} \tag{10}$$

That is:

$$\boldsymbol{m}(X_1) = k$$
$$E[\boldsymbol{m}(X_2)] = \boldsymbol{m}(X_1) + k\frac{M-k}{M} - k\frac{k}{M}$$
$$= 2k\left(1 - \frac{k}{M}\right)$$
$$\ldots$$

Now because $|X| \leq |S - C| \leq |S| \leq N$, and due to the way GBF parameters are created (see Section 2.2) BF_X and *a fortiori* BF_{X_i} should have not less than half of its bits unset, so $\boldsymbol{n}(X_i) \geq M/2$ and $k\frac{\boldsymbol{n}(X_{i-1})}{M} \geq k/2$. At the same time $\boldsymbol{m}(X_{i-1})$ is always very small compared to M. It should then obvious that $|\boldsymbol{m}(X)| \geq k$. Finally as we already explained, the probability for $\boldsymbol{m}(X)$ to be a subset of $h_*(C)$ is negligible because it is less than the probability for a single element to have all its positions in $h_*(C)$, given that we already show it has a size greater than k. □

5.2   New Proof of Security

We give a new proof of Theorem 1, that is, we show that:

$$(S, C, \mathsf{Extract}(BF_C, GBF_S)) \stackrel{\text{c}}{\equiv} (S, C, \mathsf{Extract}(BF_C, GBF_{S \cap C}))$$

We consider two cases as it is done by Dong et al. [4]: The first case is where there is a $X \subset S - C$ such that $\boldsymbol{m}(X) \subset h_*(C)$. From Theorem 2, This case happens with negligible probability. The second case is thus where there is no such X, and we show that in this case the distributions are identical by showing that any outcome of one distribution is a valid outcome of the other. Let B be an outcome of the right-hand distribution, that is, the one with $GBF_{S \cap C}$; we show how to build a GBF B' that is a valid outcome of GBF_S such that $Extract(BF_C, B') = B$. We build B' the following way: We start from B which, recall, is a Garbled Bloom Filter with all its components not in domain of C being empty. We will insert each element of $S - C$ in B, keeping components that were already set untouched. Insertion happens just as in the GBF.Build algorithm. When all elements have been inserted, the remaining components are filled with random values, just as in the end of GBF.Build. If the algorithm did not halt, the resulting B' encodes every element of $S \cap C$ (from the initial values from B) and every element of $S - C$ (that we just inserted). As a result, B' is a valid GBF_S and $Extract(BF_C, B')$ is a valid outcome for $Extract(BF_C, GBF_S)$.

   We now show that the algorithm does not halt. Recall, the building algorithm halts when an element that must be inserted only maps to positions that are not empty. Since we are in the case where no $X \subset S - C$ satisfies $\boldsymbol{m}(X) \subset h_*(C)$, there must be a position in $h_*(S - C)$ that is not in $h_*(C)$ and which is mapped by a single element $y \in S - C$. As a result if $(S - C) - \{y\}$ was inserted without halting, then the final y can be inserted without halting as well. This reasoning can be repeated to show that $(S - C) - \{y\}$ can be inserted without halting as well, and recursively $S - C$ can be inserted entirely without halting.

   Finally given an outcome B of $Extract(BF_C, GBF_S)$ one can trivially build a valid GBF B' encoding $S \cap C$ such that $Extract(BF_C, B') = B$: it suffices to fill all empty components of B with random values. As a result we have $Extract(BF_C, GBF_S) \equiv Extract(BF_C, GBF_{S \cap C})$ in our second case, and this ends the proof of Theorem 1.   □

   Note that this proof would also apply to the construction of Pinkas et al. [11].

6   Related Work

Security issues in the paper of Dong et al. [4] where identified by Rindal and Rosulek [12] and by Lambæk [8], but none of these issues apply on the protocol that we study in this paper. Indeed, [4] describes two protocol: one that aims at providing security against honest-but-curious adversaries, which is the one that is being studied in this paper, and one that aims at providing security against malicious adversaries. The issues identified in [12] and [8] only

concern the malicious-security protocol, and do not apply to the honest-but-curious-security protocol (both present the honest-but-curious-security protocol as satisfying its claimed properties).

By contrast, the issues we identify concern the security of the GBF construction. This property is invoked in the security proofs for both the honest-but-curious-security protocol and the malicious-security one, so the two protocols are affected. The issue we identify is thus different, and more general, than the ones identified in [12] and [8].

7   Conclusion

Garbled Bloom Filters are a hash structure which, however still recent, already had a significant impact on the design of secure protocols. We showed that the security analysis of Garbled Bloom Filter contains a subtle difficulty as the intuition that GBF security derives almost immediately from the security of XOR-based secret sharing is actually false. Nevertheless we show that all existing GBF constructions actually satisfy their claimed security property by providing a new, more rigorous proof. This should strengthen the confidence we can have in the GBF construction and promote a large use of it in the domain of secure protocol design.

Acknowledgements

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**(7), 422–426 (Jul 1970). https://doi.org/10.1145/362686.362692, http://doi.acm.org/10.1145/362686.362692
2. Broder, A.Z., Mitzenmacher, M.: Survey: Network Applications of Bloom Filters: A Survey. Internet Mathematics **1**(4), 485–509 (2003). https://doi.org/10.1080/15427951.2004.10129096
3. Dong, C., Chen, L.: A Fast Secure Dot Product Protocol with Application to Privacy Preserving Association Rule Mining. In: Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I. pp. 606–617 (2014)
4. Dong, C., Chen, L., Wen, Z.: When private set intersection meets big data: an efficient and scalable protocol. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013. pp. 789–800 (2013). https://doi.org/10.1145/2508859.2516701

5. Ghosh, E., Ohrimenko, O., Papadopoulos, D., Tamassia, R., Triandopoulos, N.: Zero-Knowledge Accumulators and Set Algebra. In: Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. pp. 67–100 (2016)
6. Katz, J., Lindell, Y.: Introduction to Modern Cryptography, Second Edition. CRC Press (2014)
7. Kiss, A., Liu, J., Schneider, T., Asokan, N., Pinkas, B.: Private Set Intersection for Unequal Set Sizes with Mobile Applications. IACR Cryptology ePrint Archive **2017**, 670 (2017), http://eprint.iacr.org/2017/670
8. Lambaek, M.: Breaking and Fixing Private Set Intersection Protocols. IACR Cryptology ePrint Archive **2016**, 665 (2016), http://eprint.iacr.org/2016/665
9. Lentz, M., Erdélyi, V., Aditya, P., Shi, E., Druschel, P., Bhattacharjee, B.: SDDR: Light-Weight, Secure Mobile Encounters. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014. pp. 925–940 (2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/lentz
10. Patsakis, C., Zigomitros, A., Solanas, A.: Privacy-Aware Genome Mining: Server-Assisted Protocols for Private Set Intersection and Pattern Matching. In: 28th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2015, Sao Carlos, Brazil, June 22-25, 2015. pp. 276–279 (2015). https://doi.org/10.1109/CBMS.2015.70, https://doi.org/10.1109/CBMS.2015.70
11. Pinkas, B., Schneider, T., Zohner, M.: Faster private set intersection based on OT extension. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014. pp. 797–812 (2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/pinkas
12. Rindal, P., Rosulek, M.: Improved Private Set Intersection Against Malicious Adversaries. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I. pp. 235–259 (2017)
13. Rompay, C.V., Molva, R., Önen, M.: Secure and scalable multi-user searchable encryption (2018)
14. Wang, X.S., Huang, Y., Zhao, Y., Tang, H., Wang, X., Bu, D.: Efficient Genome-Wide, Privacy-Preserving Similar Patient Query based on Private Edit Distance. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015. pp. 492–503 (2015). https://doi.org/10.1145/2810103.2813725, http://doi.acm.org/10.1145/2810103.2813725
15. Wen, Z., Dong, C.: Efficient protocols for private record linkage. In: Symposium on Applied Computing, SAC 2014, Gyeongju, Republic of Korea - March 24 - 28, 2014. pp. 1688–1694 (2014). https://doi.org/10.1145/2554850.2555001, http://doi.acm.org/10.1145/2554850.2555001
16. Zhao, Y., Chow, S.S.M.: Are you The One to Share? Secret Transfer with Access Structure. PoPETs **2017**(1), 149–169 (2017). https://doi.org/10.1515/popets-2017-0010, https://doi.org/10.1515/popets-2017-0010
17. Zheng, Q., Xu, S.: Verifiable Delegated Set Intersection Operations on Outsourced Encrypted Data. In: 2015 IEEE International Conference on Cloud Engineering, IC2E 2015, Tempe, AZ, USA, March 9-13, 2015. pp. 175–184 (2015). https://doi.org/10.1109/IC2E.2015.38, https://doi.org/10.1109/IC2E.2015.38