On the Security of the "Free-XOR" Technique*

Seung Geol Choi, Jonathan Katz, Ranjit Kumaresan, and Hong-Sheng Zhou**

Dept. of Computer Science, University of Maryland, College Park, MD 20742, USA
{sgchoi,jkatz,ranjit,hszhou}@cs.umd.edu

Abstract. Yao's *garbled-circuit approach* enables constant-round secure two-party computation of any function. In Yao's original construction, each gate in the circuit requires the parties to perform a constant number of encryptions/decryptions and to send/receive a constant number of ciphertexts. Kolesnikov and Schneider (ICALP 2008) proposed an improvement that allows XOR gates to be evaluated "for free," incurring no cryptographic operations and zero communication. Their "free-XOR" technique has proven very popular, and has been shown to improve performance of garbled-circuit protocols by up to a factor of 4.

Kolesnikov and Schneider proved security of their approach in the random oracle model, and claimed that (an unspecified variant of) correlation robustness suffices; this claim has been repeated in subsequent work, and similar ideas have since been used in other contexts. We show that the free-XOR technique *cannot* be proven secure based on correlation robustness alone; somewhat surprisingly, some form of *circular security* is also required. We propose an appropriate definition of security for hash functions capturing the necessary requirements, and prove security of the free-XOR approach when instantiated with any hash function satisfying our definition.

Our results do not impact the security of the free-XOR technique in practice, or imply an error in the free-XOR work, but instead pin down the assumptions needed to prove security.

1 Introduction

Generic protocols for secure two-party computation have been known for over 25 years [35,13]. (By "generic" we mean that the protocol is constructed by starting with a boolean or arithmetic circuit for the function of interest.) For most of that time, generic secure two-party computation was viewed as being only of theoretical interest; much effort was instead devoted to developing more efficient, "tailored" protocols for specific functions of interest.

In recent years, however, a number of works have shown that generic protocols for secure two-party computation may be much more attractive than previously thought. This line of work was initiated by Fairplay [29], which gave an implementation of Yao's garbled-circuit protocol [35] secure in the semi-honest setting.

* This work was supported in part by NSF awards #0447075 and #1111599, and by DARPA.

** Supported by an NSF CI postdoctoral fellowship.

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 39–53, 2012.

Subsequent works showed improvements in the scalability, efficiency, and usability of garbled circuits [17,24,19,20], extended the garbled-circuit technique to give implementations in the malicious setting [28,33,34], and explored alternatives to the garbled-circuit approach [23,17,11,31].

As secure computation moves from theory to practice, even small improvements can have a significant effect. (Three factor-of-2 improvements can reduce the time from, say, 1 minute to under 8 seconds.) Indeed, several such improvements have been proposed for the garbled-circuit approach: e.g., the *point-and-permute* technique [29] that reduces the circuit evaluator's work (per gate) from four decryptions to one, or *garbled-row reduction* [30,33] that reduces the number of ciphertexts transmitted per garbled gate from four to three.

It is in this spirit that Kolesnikov and Schneider introduced their very influential "free-XOR" approach [26] for improving the efficiency of garbled-circuit constructions. (The free-XOR optimization is compatible with both the point-and-permute technique and garbled-row reduction.) Yao's original construction requires a garbled gate for each boolean gate in the circuit of the function being computed. The free-XOR technique allows XOR gates in the underlying circuit to be evaluated "for free," without the need to construct a corresponding garbled gate. (We defer the technical details to Section 2.2.) XOR gates in the underlying circuit therefore incur no communication cost or cryptographic operations. Because of this, as documented in [26,25,33], it is worth investing the effort to minimize the number of non-XOR gates in the underlying circuit (even if the total number of gates is increased); this results in roughly a 40% overall efficiency improvement for "typical" circuits [33]. For some circuits (e.g., basic arithmetic operations, universal circuits) a factor-of-4 improvement is observed [26,25]. Nowadays, all implementations of garbled-circuit protocols use the free-XOR idea to improve performance [33,17,34,24,19,20].

1.1 Security of the Free-XOR Technique?

Given the popularity of the free-XOR technique, it is natural to ask what are the necessary assumptions based on which it can be proven secure.[1] The free-XOR approach relies on a cryptographic hash function H. Kolesnikov and Schneider [26] gave a proof of security for the free-XOR technique when H is modeled as a random oracle, and claimed that (a variant of) *correlation robustness* [21,14] would be sufficient; this claim has been repeated in several subsequent works [33,3,7]. (Informally, correlation robustness implies that $H(k_1 \oplus R), \ldots, H(k_t \oplus R)$ are all pseudorandom, even given $k_1, \ldots, k_t$, when R is chosen at random. In the context of the free-XOR technique we must consider hash functions taking two inputs. Formal definitions are given in Section 2.3.) Correlation robustness is a relatively mild assumption, and has the advantage

[1] It may be interesting to recall here that XOR gates are also "free" when using the GMW approach to secure two-party computation [13]. In that setting, no additional assumptions are needed.

relative to the random-oracle model of being (potentially) falsifiable. Moreover, correlation robustness is already required by existing protocols for oblivious-transfer extension [21], which are used in current efficient implementations of secure two-party computation.

Our Results. It is unclear exactly what variant of correlation robustness is needed to prove security of the free-XOR approach, and Kolesnikov and Schneider (as well as subsequent researchers relying on their result) have left this question unanswered. We show here that the natural variant of correlation robustness (for hash functions taking two inputs instead of one) is *not* sufficient. We describe where the obvious attempt to prove security fails, and moreover show an explicit counterexample (in the random-oracle model) of a correlation-robust hash function H for which the free-XOR approach is demonstrably insecure.

We observe that the difficulty is due to a previously unnoticed *circularity* in the free-XOR construction: in essence, the issue is that $H(k_1 \oplus R)$ is used to encrypt both k_2 and $k_2 \oplus R$. (The actual issue is more involved, and depends on the details of the free-XOR approach; see Section 3.) We thus define a notion of *circular* correlation robustness, and show that any hash function satisfying this definition can be used to securely instantiate the free-XOR technique. Our definition is falsifiable, and is still weaker than modeling H as a random oracle. Our work can be viewed as following the line of research suggested in [10] whose goal is to formalize, and show usefulness of, various concrete properties satisfied by a random oracle.

Besides the original work of Kolesnikov and Schneider, our results also impact security claims made in two other recent papers. Nielsen and Orlandi [32] use an idea similar to that used in the free-XOR approach to construct a (new) protocol for two-party computation secure against malicious adversaries. They, too, prove security in the random-oracle model but claim that correlation robustness suffices; their construction appears to have the same issues with circularity that the free-XOR technique has. Applebaum et al. [3] define a notion of security against passive related-key attacks for encryption schemes, and claim that encryption schemes satisfying this notion can be used to securely instantiate the free-XOR approach (see [3, Section 1.1.2]). However, their definition of related-key attacks does not take into account any notion of circular security, which appears to be necessary for the free-XOR technique to be sound. We conjecture that our new definition of circular correlation robustness suffices to prove security in each of the above works.

We do not claim that our work has any impact on the security of the free-XOR technique (or the protocols of [32,3]) in practice; in most cases, protocol implementors seem content to assume the random-oracle model anyway. Nevertheless, it is important to understand the precise assumptions needed to prove these protocols secure. We also do not claim any explicit error in the work of Kolesnikov and Schneider [26], as they only say that *some* variant of correlation robustness should suffice. Our work pins down exactly what variant of correlation robustness is necessary.

1.2 Related Work

The notion of correlation robustness was introduced by Ishai et al. [21], and has been used in several other works since then [16,22,32,33]. Applebaum et al. [3] and Goyal et al. [14] further study the notion, explore various definitions, and show connections to security against related-key attacks [5,12,4]. To the best of our knowledge, none of the previous definitions of correlation robustness given in the literature suffice to prove security of the free-XOR technique.

As mentioned above, we define a notion of security for hash functions that blends correlation robustness and circular security. The latter notion, as well as the more general notion of key-dependent-message security, has seen a significant amount of attention recently [6,15,18,2,8,1,9].

1.3 Organization

We review Yao's garbled-circuit construction, and the free-XOR modification of it, in Section 2. In that section we also define a notion of correlation robustness that is syntactically suitable for trying to prove security of the free-XOR approach. In Section 3 we explain where a reductionist proof of security for the free-XOR approach fails when trying to base security on correlation robustness alone. We then demonstrate that *no* proof of security is possible by showing an example of a correlation-robust hash function for which the free-XOR approach is demonstrably insecure. This motivates our definition of a stronger notion of security for hash functions in Section 4, one that we show suffices for proving security of the free-XOR technique.

2 Preliminaries

2.1 Yao's Garbled Circuit Construction

Yao's garbled-circuit approach [35], in combination with any oblivious-transfer protocol, yields a constant-round protocol for two-party computation with security against semi-honest parties. We review only those aspects of the construction needed to understand our results; for further details, we refer to [26,27].

Fix a boolean circuit C known to both parties. (For simplicity, we assume the circuit C outputs a single bit; the protocol can be easily extended to handle multi-bit outputs.) One party, the garbled-circuit generator, prepares a garbled version of the circuit as follows. First, two random keys w_i^0, w_i^1 are associated with each wire i in the circuit; key w_i^0 corresponds to the value '0' on wire i, while w_i^1 corresponds to the value '1'. For each wire i, a random bit π_i is also chosen; key w_i^b is assigned the label $\lambda_i^b = b \oplus \pi_i$. For each gate $g : \{0,1\}^2 \to \{0,1\}$ in the circuit, with input wires i, j and output wire k, the circuit generator constructs a "garbled gate" that will enable the other party to recover $w_k^{g(b_i,b_j)}$ (and its label)

from $w_i^{b_i}$ and $w_j^{b_j}$ (and their corresponding labels). The garbled gate consists of the four ciphertexts:

$$\mathsf{Enc}^g_{w_i^{\pi_i}, w_j^{\pi_j}}\left(w_k^{g(\pi_i,\pi_j)} \| g(\pi_i,\pi_j)\oplus\pi_k\right) \tag{1}$$

$$\mathsf{Enc}^g_{w_i^{\pi_i}, w_j^{1\oplus\pi_j}}\left(w_k^{g(\pi_i,1\oplus\pi_j)} \| g(\pi_i,1\oplus\pi_j)\oplus\pi_k\right) \tag{2}$$

$$\mathsf{Enc}^g_{w_i^{1\oplus\pi_i}, w_j^{\pi_j}}\left(w_k^{g(1\oplus\pi_i,\pi_j)} \| g(1\oplus\pi_i,\pi_j)\oplus\pi_k\right) \tag{3}$$

$$\mathsf{Enc}^g_{w_i^{1\oplus\pi_i}, w_j^{1\oplus\pi_j}}\left(w_k^{g(1\oplus\pi_i,1\oplus\pi_j)} \| g(1\oplus\pi_i,1\oplus\pi_j)\oplus\pi_k\right), \tag{4}$$

in that order. (In the above, we use $\mathsf{Enc}^g_{w,w'}(\cdot)$ to denote encryption under the two keys w and w' that may also depend on the gate number g. The exact details of the encryption will be specified in the next section, but for concreteness the reader can for now think of it as being instantiated by $\mathsf{Enc}^g_{w,w'}(m) = \mathsf{Enc}_w(\mathsf{Enc}_{w'}(m))$ with the gate number being ignored. We use here the point-and-permute technique so that the circuit evaluator only needs to decrypt a single ciphertext per garbled gate.) To evaluate this garbled gate, the circuit evaluator who holds $w_i^{b_i} \| \lambda_i^{b_i}$ and $w_j^{b_j} \| \lambda_j^{b_j}$ uses those keys to decrypt the ciphertext at position $\lambda_i^{b_i}, \lambda_j^{b_j}$ of the above array; this will recover $w_k^{g(b_i,b_j)} \| \lambda_k^{g(b_i,b_j)}$, where $\lambda_k^{g(b_i,b_j)} = g(b_i,b_j)\oplus\pi_k$ as required.

Let $i_1, \ldots, i_\ell$ denote the input wires of the circuit. With garbled gates constructed as above for each gate of the circuit (and transmitted to the circuit evaluator), we see that given keys $w_{i_1}^{b_1}, \ldots, w_{i_\ell}^{b_\ell}$ (and their corresponding labels) for the input wires, the circuit evaluator can inductively compute a key (and its label) for the output wire. The keys for input wires belonging to the circuit generator can simply be transmitted to the circuit evaluator along with the garbled gates; the keys for input wires belonging to the circuit evaluator are obtained by the circuit evaluator using oblivious transfer (OT). If the circuit generator also sends π_o for the output wire o, the circuit evaluator can obtain the correct boolean output of the circuit on the given inputs.

The above thus defines a protocol for two-party computation in the OT-hybrid model. If encryption is instantiated via $\mathsf{Enc}^g_{w,w'}(m) = \mathsf{Enc}_w(\mathsf{Enc}_{w'}(m))$ and Enc is a CPA-secure symmetric-key encryption scheme, the protocol is secure against semi-honest adversaries [35,27].

2.2 The Free-XOR Technique

Kolesnikov and Schneider [26] suggested that instead of choosing the keys w_i^0, w_i^1 for each wire i independently at random, one could instead (1) choose a global random value R, (2) choose w_i^0 uniformly and independently at random for every wire i that is not the output of an XOR gate, and (3) set $w_i^1 = w_i^0 \oplus R$. Each such wire is also assigned a random bit π_i as before. If k is the output wire of an XOR gate with input wires i, j (whose keys have already been defined), then the keys for wire k are set to be $w_k^0 = w_i^0 \oplus w_j^0$ and $w_k^1 = w_k^0 \oplus R$; also, π_k

is set to be $\pi_k = \pi_i \oplus \pi_j$. If keys are chosen this way, then for any XOR gate as above the circuit evaluator holding $w_i^{b_i} \| \lambda_i^{b_i}$ and $w_j^{b_j} \| \lambda_j^{b_j}$ can simply compute $w_k^{b_i \oplus b_j} = w_i^{b_i} \oplus w_j^{b_j}$ and $\lambda_k^{b_i \oplus b_j} = \lambda_i^{b_i} \oplus \lambda_j^{b_j}$; this is correct since $w_i^{b_i} = w_i^0 \oplus b_i R$ (and similarly for $w_j^{b_j}$), where the notation $b_i R$ evaluates to $0^{|R|}$ if $b_i = 0$ or to R otherwise, and thus

$$w_i^{b_i} \oplus w_j^{b_j} = w_i^0 \oplus w_j^0 \oplus (b_i \oplus b_j)R = w_k^0 \oplus (b_i \oplus b_j)R = w_k^{b_i \oplus b_j}$$

and

$$\lambda_i^{b_i} \oplus \lambda_j^{b_j} = (b_i \oplus \pi_i) \oplus (b_j \oplus \pi_j) = (b_i \oplus b_j) \oplus \pi_k = \lambda_k^{b_i \oplus b_j}.$$

Note that by doing so, XOR gates incur no communication and require no cryptographic operations by either party. For the remaining (non-XOR) gates in the circuit, the circuit generator prepares garbled gates as in the previous section.

As previously, the above defines a protocol for secure two-party computation in the OT-hybrid model. Kolesnikov and Schneider suggest to implement encryption using a cryptographic hash function H as follows:

$$\mathsf{Enc}^g_{w,w'}(m) = H(w \| w' \| g) \oplus m.$$

When encryption is instantiated in this way, Kolesnikov and Schneider prove security of their protocol, for semi-honest adversaries, when H is modeled as a random oracle. They also claimed that security would hold if H satisfies some "variant" of correlation robustness. While they did not specify precisely what variant of correlation robustness is needed, a natural approach would be that they require the (joint) pseudorandomness of $H(w \| w' \| g)$, $H(w \oplus R \| w' \| g)$, $H(w \| w' \oplus R \| g)$, $H(w \oplus R \| w' \oplus R \| g)$ for w, w', R chosen at random. We discuss this issue further in the following section.

2.3 Correlation-Robust Hash Functions

As noted at the end of the previous section, Kolesnikov and Schneider claim that some variant of correlation robustness would be sufficient to prove security of the free-XOR construction. Let $H = \{H_n : \{0,1\}^{\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}\}$ be a family of hash functions, where for simplicity we write H instead of H_n when the security parameter n is understood. Correlation robustness was defined by Ishai et al. [21] as follows:

Definition 1. *H is* correlation robust *if for any polynomial $p(\cdot)$ and any non-uniform polynomial-time distinguisher $\mathcal{A}$, the following is negligible in the security parameter n:*

$$\Big| \Pr_{w_1,\ldots,w_p,R \leftarrow \{0,1\}^{\ell_{in}(n)}} \Big[\mathcal{A}\big(w_1, \ldots, w_p, H(w_1 \oplus R), \ldots, H(w_p \oplus R)\big) = 1 \Big]$$

$$- \Pr_{w_1,\ldots,w_p \leftarrow \{0,1\}^{\ell_{in}(n)}, u_1,\ldots,u_p \leftarrow \{0,1\}^{\ell_{out}(n)}} \Big[\mathcal{A}\big(w_1, \ldots, w_p, u_1, \ldots, u_p\big) = 1 \Big] \Big|,$$

where $p = p(n)$.

In the context of the free-XOR technique as defined by Kolesnikov and Schneider, an appropriate definition of correlation robustness needs to at least capture the security requirement (informally) that given any pair of keys $w_i^{b_i}, w_j^{b_j}$ for some garbled gate constructed as in Equations (1)–(4), with $\mathsf{Enc}^g_{w,w'}(m) = H(w\|w'\|g)\oplus m$, it should be possible to decrypt only one row while the others remain hidden. Since the hash function H now takes three inputs, the definition of correlation robustness needs to be modified appropriately. Moreover, for the free-XOR approach it appears necessary to allow w_i to take on arbitrary values[2] rather than being chosen uniformly and independently at random; roughly, this is because in the free-XOR construction we have $w_k^0 = w_i^0 \oplus w_j^0$ when k is the output wire of an XOR gate with input wires i, j, and so w_i^0, w_j^0, w_k^0 are not independent. We capture these requirements in the following definition.

Definition 2. *$H : \{0,1\}^{3\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}$ is* (weakly) 2-correlation robust *if for all polynomials $p(\cdot)$ the distribution ensemble*

$$\left\{ \begin{array}{l} R \leftarrow \{0,1\}^{\ell_{in}(n)} : \\ \\ \quad H(w_1 \oplus R \| w_1' \| 1), H(w_1 \| w_1' \oplus R \| 1), H(w_1 \oplus R \| w_1' \oplus R \| 1) \\ \qquad\qquad \vdots \\ \quad H(w_p \oplus R \| w_p' \| p), H(w_p \| w_p' \oplus R \| p), H(w_p \oplus R \| w_p' \oplus R \| p) \end{array} \right\}_{\substack{w_1,\ldots,w_p \in \{0,1\}^{\ell_{in}(n)} \\ w_1',\ldots,w_p' \in \{0,1\}^{\ell_{in}(n)}}}$$

is computationally indistinguishable from the uniform distribution over $\{0,1\}^{3p\cdot\ell_{out}(n)}$. (In both cases, $p = p(n)$.)

Simplified to the case $p = 1$ with w_1, w_1' chosen uniformly and independently (and ignoring the last input to H), the definition requires that the values

$$H(w_1 \oplus R \| w_1'), H(w_1 \| w_1' \oplus R), H(w_1 \oplus R \| w_1' \oplus R)$$

be jointly pseudorandom even given w_1, w_1'. Note that this is equivalent to, say, requiring that

$$H(w_1 \| w_1'), H(w_1 \oplus R \| w_1' \oplus R), H(w_1 \| w_1' \oplus R)$$

be jointly pseudorandom given $w_1 \oplus R, w_1'$, and thus may appear to capture the requirements necessary for proving the free-XOR technique secure.

It will be more convenient to rephrase the above as an oracle-based definition, and this also provides a point of departure for the definition we will propose in Section 4. (In fact, the oracle-based definition we give is stronger than Definition 2 as it allows the adversary to adaptively choose w_i, w_i' based on previous outputs of H. But see footnote 2.) Fixing some H, define oracles $\mathsf{Cor}_R(\cdot,\cdot,\cdot)$ and $\mathsf{Rand}(\cdot,\cdot,\cdot)$ as follows:

[2] We will show impossibility of proving security based on correlation robustness alone. Thus, a stronger definition of correlation robustness only strengthens that result.

– $\mathsf{Cor}_R(w, w', g)$: output $H(w\|w'\oplus R\|g)$, $H(w\oplus R\|w'\|g)$, and $H(w\oplus R\|w'\oplus R\|g)$.
– $\mathsf{Rand}(w, w', g)$: if this input was queried before then return the answer given previously. Otherwise choose $u \leftarrow \{0,1\}^{3\cdot\ell_{out}(n)}$ and return u.

We now have the following definition:

Definition 3. *H is* 2-correlation robust *if for all non-uniform polynomial-time distinguishers $\mathcal{A}$ the following is negligible:*

$$\left|\Pr[R\leftarrow\{0,1\}^{\ell_{in}(n)} : \mathcal{A}^{\mathsf{Cor}_R(\cdot)}(1^n) = 1] - \Pr[\mathcal{A}^{\mathsf{Rand}(\cdot)}(1^n) = 1]\right|.$$

3 Insufficiency of Correlation Robustness

In this section, we show that 2-correlation robustness is not enough to prove security of the free-XOR technique. We start by describing where the natural attempt to prove security (following, e.g., the proof of [27]) fails. We then show a construction (in the random-oracle model) of a hash function H that satisfies Definition 3 but for which the free-XOR approach is demonstrably insecure when instantiated using H.

3.1 Where the Reduction Fails

Consider the case where the circuit consists of just a single AND gate, with input wires 1 and 2 (belonging to the circuit generator and evaluator, respectively) and output wire 3. Say the circuit evaluator has input 0 and so receives key w_2^0; assume for concreteness that the circuit generator has input 0 as well and so the circuit evaluator is also given key w_1^0. (The circuit evaluator will also be given the corresponding labels, but these can be left implicit in what follows.) The garbled gate consists of the values

$$\begin{aligned}
&H(w_1^0\|w_2^0\|1) \oplus (w_3^0\|0)\\
&H(w_1^0\|w_2^0\oplus R\|1) \oplus (w_3^0\|0)\\
&H(w_1^0\oplus R\|w_2^0\|1) \oplus (w_3^0\|0)\\
&H(w_1^0\oplus R\|w_2^0\oplus R\|1) \oplus \big((w_3^0\oplus R)\|1\big)
\end{aligned}$$

in some permuted order, for some random value R unknown to the circuit evaluator. (Recall that $w_i^1 = w_i^0\oplus R$ for all i, by construction, when using the free-XOR approach.) The evaluator will be able to decrypt the first row, above, to learn the output; it should not, however, be able to learn any information about the remaining three rows. (In particular, it should not learn whether the other party had input 0 or 1.) The natural way to try to prove security of the above is to argue that the remaining rows are pseudorandom, by reduction to the 2-correlation robustness of H. In the reduction, we would have an adversary $\mathcal{A}$ given access to an oracle $\mathcal{O}$ that is either Cor_R (for a random R) or Rand. The adversary $\mathcal{A}$ can choose random w_1^0, w_2^0, and then query $\mathcal{O}(w_1^0, w_2^0, 1)$ to obtain three values h_1, h_2, h_3 that are either completely random or equal to $H(w_1^0\|w_2^0\oplus R\|1)$,

$H(w_1^0 \oplus R \| w_2^0 \| 1)$, and $H(w_1^0 \oplus R \| w_2^0 \oplus R \| 1)$. But $\mathcal{A}$ cannot complete the simulation, since it has no way to compute values of the form

$$h_1 \oplus (w_3^0 \| 0), \quad h_2 \oplus (w_3^0 \| 0), \quad h_3 \oplus \Big((w_3^0 \oplus R) \| 1 \Big)$$

(since $\mathcal{A}$ does not know R) as would be necessary to simulate the remaining three rows of the garbled gate in case $\mathcal{O} = \mathsf{Cor}_R$.

We show in the next section that this is not just a failure of this particular proof approach, since we can construct a hash function H that satisfies Definition 3 yet for which the free-XOR methodology is demonstrably insecure when instantiated using H.

3.2 A Counter-Example

For simplicity, we fix a value of the security parameter n. Assume further that the last input to H (i.e., the gate index g) is represented using n bits. We construct a pair of oracles $H : \{0,1\}^{3n} \to \{0,1\}^{n+1}$ and $\mathsf{Break} : \{0,1\}^{6n+3} \to \{0,1\}^n$ such that:

- H satisfies Definition 3, even if the distinguisher $\mathcal{A}$ is given oracle access to both H and Break.
- The free-XOR methodology is demonstrably insecure when instantiated using H, against an adversary given oracle access to both H and Break.

Thus, we rule out a fully black-box reduction of the security of the free-XOR technique to 2-correlation robustness.

Let $H : \{0,1\}^{3n} \to \{0,1\}^{n+1}$ be a random function, and define Break as follows:

> $\underline{\mathsf{Break}(w \| w' \| g \| z_1 \| z_2 \| z_3)}$: If there exists $r \in \{0,1\}^n$ such that
>
> $z_1 = H(w \| w' \oplus r \| g)$, $z_2 = H(w \oplus r \| w' \| g)$, and $z_3 = H(w \oplus r \| w' \oplus r \| g) \oplus (r \| 0)$,
>
> then output r (if multiple values of r satisfy the above, take the lexicographically smallest one); otherwise, output $\perp$.

We now prove the above claims.

Lemma 1. *H is 2-correlation robust, even when the distinguisher is given oracle access to both H and Break.*

Proof (Sketch). Fix a polynomial-time distinguisher $\mathcal{A}$ who is given access to $H, \mathsf{Break}, \mathcal{O}$ where either $\mathcal{O} = \mathsf{Cor}_R$ (for random $R \in \{0,1\}^n$) or $\mathcal{O} = \mathsf{Rand}$. Without loss of generality, we assume that $\mathcal{A}$ does not repeat queries to $\mathcal{O}$. When $\mathcal{O} = \mathsf{Rand}$, every query to $\mathcal{O}$ is answered with a string of length $3 \cdot (n+1)$ that is uniform and independent of $\mathcal{A}$'s view. When $\mathcal{O} = \mathsf{Cor}_R$, every query $\mathcal{O}(w, w', g)$ is answered with a string of length $3 \cdot (n+1)$ that is uniform and independent of $\mathcal{A}$'s view unless one of the following is true:

- $\mathcal{A}$ at some point queries $\mathcal{O}(\tilde{w}, \tilde{w}', g)$ with $\tilde{w} \oplus w = R$ or $\tilde{w}' \oplus w' = R$ (or both).
- $\mathcal{A}$ at some point queries $H(\tilde{w} \| \tilde{w}' \| g)$ with $\tilde{w} \oplus w = R$ or $\tilde{w}' \oplus w' = R$ (or both).
- $\mathcal{A}$ at some point queries $\mathsf{Break}(w \| w' \| g \| z_1 \| z_2 \| z_3)$ where it holds that $R \| 0 = z_3 \oplus H(w \oplus R \| w' \oplus R \| g)$.

Since R is chosen uniformly from $\{0,1\}^n$, the probability that $\mathcal{A}$ makes any queries of the above form is negligible.

Lemma 2. *The free-XOR construction, when instantiated using H, is not secure against a semi-honest adversary (with oracle access to H and Break) who corrupts the circuit evaluator.*

Proof. We show that a semi-honest adversary can recover R with high probability. Since a semi-honest adversary can (legitimately) recover one key per wire by following the protocol, knowledge of R allows the adversary to recover *both* keys for every wire in the circuit; thus, this suffices to show that the construction is insecure.

Assume the first gate in the circuit is an AND gate with input wires 1 and 2 (belonging to the circuit generator and evaluator, respectively) and output wire 3. Say the circuit evaluator has input 0 and so receives key w_2^0; assume for concreteness that the circuit generator has input 0 as well and so the circuit evaluator is also given key w_1^0. With constant probability we have $\pi_1 = \pi_2 = \pi_3 = 0$, and in that case the garbled gate consists of the values

$$\begin{aligned}
c_{00} &= H(w_1^0 \| w_2^0 \| 1) \oplus (w_3^0 \| 0) \\
c_{01} &= H(w_1^0 \| w_2^0 \oplus R \| 1) \oplus (w_3^0 \| 0) \\
c_{10} &= H(w_1^0 \oplus R \| w_2^0 \| 1) \oplus (w_3^0 \| 0) \\
c_{11} &= H(w_1^0 \oplus R \| w_2^0 \oplus R \| 1) \oplus \left((w_3^0 \oplus R) \| 1 \right).
\end{aligned}$$

The circuit evaluator can compute w_3^0 from c_{00} (as directed by the protocol). It then computes

$$\begin{aligned}
z_1 &= c_{01} \oplus (w_3^0 \| 0) \\
z_2 &= c_{10} \oplus (w_3^0 \| 0) \\
z_3 &= c_{11} \oplus (w_3^0 \| 1)
\end{aligned}$$

and queries $\mathsf{Break}(w_1^0 \| w_2^0 \| 1 \| z_1 \| z_2 \| z_3)$. If the answer is some value $R' \neq \perp$ then with overwhelming probability it holds that $R' = R$. (Correctness of R can also be verified by looking at a second garbled gate with known inputs.)

4 Proving Security of the Free-XOR Approach

The essence of the problem(s) described in the previous section is that there is a previously unnoticed *circularity* in the free-XOR approach, since in the general case both $H(w_1 \| w_2 \| g) \oplus w_3$ and $H(w_1 \oplus R \| w_2 \oplus R \| g) \oplus (w_3 \oplus R)$ are revealed

to the adversary. (Recall that R is the hidden secret here.) In this section, we introduce a new security definition that explicitly takes this circularity into account, and show that this definition suffices to prove security of the free-XOR approach.

Fix some function $H : \{0,1\}^{3\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}$. We define an oracle Circ_R as follows:[3]

- $\mathsf{Circ}_R(w, w', g, b_1, b_2, b_3)$ outputs $H(w \oplus b_1 R \| w' \oplus b_2 R \| g) \oplus b_3 R$.

To see the connection with the previous definition (in the context of correlation robustness), note that oracle $\mathsf{Cor}_R(w, w', g)$ defined previously outputs $\mathsf{Circ}_R(w, w', g, 0, 1, 0)$, $\mathsf{Circ}_R(w, w', g, 1, 0, 0)$, and $\mathsf{Circ}_R(w, w', g, 1, 1, 0)$; i.e., b_3 was fixed to 0 there. The possibility of $b_3 = 1$ is exactly what models circularity involving R.

Corresponding to the above we define an oracle Rand in a way analogous to before:

- $\mathsf{Rand}(w, w', g, b_1, b_2, b_3)$: if this input was queried before then return the answer given previously. Otherwise choose $u \leftarrow \{0,1\}^{\ell_{out}(n)}$ and return u.

In our new definition of security for H, we are going to require that oracles Circ_R (for random R) and Rand be indistinguishable. This cannot possibly be true, however, unless we rule out some trivial queries that can be used to distinguish them. Let $\mathcal{O}$ be the oracle to which a distinguisher is given access, where either $\mathcal{O} = \mathsf{Circ}_R$ or $\mathcal{O} = \mathsf{Rand}$. We must restrict the distinguisher as follows: (1) it is not allowed to make any query of the form $\mathcal{O}(w, w', g, 0, 0, b_3)$ (since it can compute $H(w \| w' \| g)$ on its own) and (2) it is not allowed to query *both* $\mathcal{O}(w, w', g, b_1, b_2, 0)$ and $\mathcal{O}(w, w', g, b_1, b_2, 1)$ for any values w, w', g, b_1, b_2 (since that would allow it to trivially recover R). We say that any distinguisher respecting these restrictions makes *legal queries*.

With this in place we can now define our notion of circular 2-correlation robustness.

Definition 4. *H is* circular 2-correlation robust *if for any non-uniform polynomial-time distinguisher $\mathcal{A}$ making legal queries to its oracle, the following is negligible:*

$$\left| \Pr[R \leftarrow \{0,1\}^{\ell_{in}(n)} : \mathcal{A}^{\mathsf{Circ}_R(\cdot)}(1^n) = 1] - \Pr[\mathcal{A}^{\mathsf{Rand}(\cdot)}(1^n) = 1] \right|.$$

Next, we show that this notion of security suffices to prove security of the free-XOR approach:

Theorem 1. *Consider the protocol described in Section 2.2 for two-party computation in the OT-hybrid model. If H as used there is circular 2-correlation robust, then the resulting protocol is secure against a semi-honest adversary.*

[3] Here, we slightly abuse the notation bR so that it evaluates to $0^{\ell_{out}(n)}$ if $b = 0$ or $R \| 0^{\ell_{out}(n) - \ell_{in}(n)}$ otherwise.

Proof. The case where the circuit generator is corrupted is trivial. Therefore, we consider corruption of the circuit evaluator B. We describe a simulator who is given the input of B and the output $z \in \{0,1\}$ of evaluating the function, and must provide B with a simulated garbled circuit that is indistinguishable from the actual one that would be sent during a real execution of the protocol. The high-level idea is exactly the same as in [27]; the crucial difference is that we reduce to circular 2-correlation robustness of H.

The simulator proceeds as follows:

1. For each wire i in the circuit that is not an output wire of an XOR gate, choose $w_i \leftarrow \{0,1\}^n$ and $\lambda_i \leftarrow \{0,1\}$.
2. For each wire k in the circuit that is the output wire of an XOR gate with input wires i, j (for which $w_i, \lambda_i, w_j, \lambda_j$ have already been defined), set $w_k = w_i \oplus w_j$ and $\lambda_k = \lambda_i \oplus \lambda_j$.
3. For each non-XOR gate g in the circuit with input wires i, j and output wire k, output the four ciphertexts c_{00}, c_{01}, c_{10}, and c_{11} as the corresponding garbled gate, where $c_{\lambda_i \lambda_j} = H(w_i \| w_j \| g) \oplus (w_k \| \lambda_k)$ and the remaining three ciphertexts are uniform strings of length $n+1$.
4. For the output wire o, set $\pi_o = \lambda_o \oplus z$.

Say $i_1, \ldots, i_\ell$ are the input wires of the circuit belonging to the circuit generator, and $j_1, \ldots, j_{\ell'}$ are the input wires belonging to the circuit evaluator. The simulator gives to B the values $w_{j_1}, \ldots, w_{j_{\ell'}}$ (as if they came from the calls to the OT functionality), and the simulated communication that includes (1) the keys $w_{i_1}, \ldots, w_{i_\ell}$, (2) the garbled gate for each non-XOR gate in the circuit, and (3) the value π_o corresponding to the output wire.

We claim that the simulated view is indistinguishable from the real-world execution of the protocol. Assume there is an adversary B who can distinguish the two distributions when the inputs to the parties are x and y, respectively, and the output is z. We show an adversary $\mathcal{A}$ who breaks the circular 2-correlation robustness of H. Given access to an oracle $\mathcal{O}$ (that is either Circ or Rand), adversary $\mathcal{A}$ does as follows:

1. For each wire i in the circuit that is not an output wire of an XOR gate, choose $w_i \leftarrow \{0,1\}^n$ and $\lambda_i \leftarrow \{0,1\}$.
2. For each wire k in the circuit that is the output wire of an XOR gate with input wires i, j (for which $w_i, \lambda_i, w_j, \lambda_j$ have already been defined), set $w_k = w_i \oplus w_j$ and $\lambda_k = \lambda_i \oplus \lambda_j$.
3. For each wire i, let $b_i \in \{0,1\}$ be the actual value on wire i; this can be determined since $\mathcal{A}$ knows the actual input (x, y) to the circuit. Set $w_i^{b_i} = w_i$, $\pi_i = \lambda_i \oplus b_i$, $\lambda_i^0 = \pi_i$, and $\lambda_i^1 = 1 \oplus \pi_i$ (i.e., only $w_i^{1-b_i}$s are left undefined).
4. For each non-XOR gate g in the circuit with input wires i, j and output wire k, output the four ciphertexts c_{00}, c_{01}, c_{10}, and c_{11} as the corresponding garbled gate, where these ciphertexts are constructed as follows:
   - $c_{\lambda_i^{b_i} \lambda_j^{b_j}} = H(w_i^{b_i} \| w_j^{b_j} \| g) \oplus (w_k^{b_k} \| \lambda_k^{b_k})$.

– For $(\lambda_i^{\beta_i}, \lambda_j^{\beta_j}) \in \{0,1\}^2$ with $(\beta_i, \beta_j) \neq (b_i, b_j)$, query

$$h_{\beta_i,\beta_j} = \mathcal{O}(w_i, w_j, g, \beta_i \oplus b_i, \beta_j \oplus b_j, g(\beta_i, \beta_j) \oplus b_k),$$

and set $c_{\lambda_i^{\beta_i} \lambda_j^{\beta_j}} = h_{\beta_i,\beta_j} \oplus (w_k^{b_k} \| \lambda_k^{g(\beta_i,\beta_j)})$.

5. For the output wire o, set $\pi_o = \lambda_o \oplus z$ (where z is the known output of the circuit).

$\mathcal{A}$ gives to B the values $w_{j_1}, \ldots, w_{j_{\ell'}}$ (as if they came from the calls to the OT functionality), and (1) the keys $w_{i_1}, \ldots, w_{i_\ell}$, (2) the garbled gate for each non-XOR gate in the circuit, and (3) the value π_o corresponding to the output wire. Finally, $\mathcal{A}$ outputs whatever B outputs. It is easy to see that $\mathcal{A}$ makes legal queries to its oracle. Furthermore, it is also easy to see that if $\mathcal{O} = \mathsf{Circ}$ the view of B is identically distributed to its view in the real execution of the protocol on the given inputs, whereas if $\mathcal{O} = \mathsf{Rand}$ the view of B is distributed identically to the output of the simulator described previously. This completes the proof.

5 Conclusion

The free-XOR technique has been extremely influential, and it is currently used in all implementations of the garbled-circuit technique because of the speedup that it gives. It was previously known that this approach is secure in the random-oracle model; it was also claimed that *some* variant of correlation robustness would suffice to prove security, but the exact notion of correlation robustness needed was left unspecified. In this work, we explore this question. We show that the natural variant of correlation robustness (extended to handle hash functions taking several inputs, rather than one input) is not sufficient, and identify a previously unnoticed circularity in the free-XOR construction that causes the difficulty. We are thus motivated to propose a new, stronger notion of correlation robustness, and we prove that this notion suffices.

Several intriguing open questions remain. First, is there some variant of the free-XOR approach that does not rely on any assumptions beyond CPA-secure encryption (which is all that is needed to prove security of classical garbled-circuit protocols in the OT-hybrid world)? Alternately, can our definition of circular 2-correlation robustness be realized from standard cryptographic assumptions?

Acknowledgments. The second author would like to thank Vlad Kolesnikov and Thomas Schneider for their feedback and encouragement.

References

1. Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic Agility and Its Relation to Circular Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 403–422. Springer, Heidelberg (2010)

2. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
3. Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: 2nd Symposium on Innovations in Computer Science (ICS), pp. 45–60. Tsinghua University Press (2011)
4. Bellare, M., Cash, D.: Pseudorandom Functions and Permutations Provably Secure against Related-Key Attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)
5. Bellare, M., Kohno, T.: A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)
6. Black, J., Rogaway, P., Shrimpton, T.: Encryption-Scheme Security in the Presence of Key-Dependent Messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)
7. Blanton, M., Gasti, P.: Secure and Efficient Protocols for Iris and Fingerprint Identification. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 190–209. Springer, Heidelberg (2011)
8. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
9. Brakerski, Z., Goldwasser, S.: Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability (or: Quadratic residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)
10. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (2004)
11. Choi, S., Hwang, K.-W., Katz, J., Malkin, T., Rubenstein, D.: Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces. In: Topics in Cryptology — Cryptographers' Track (CT-RSA 2012) (to appear, 2012)
12. Goldenberg, D., Liskov, M.: On Related-Secret Pseudorandomness. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 255–272. Springer, Heidelberg (2010)
13. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game, or a completeness theorem for protocols with honest majority. In: 19th Annual ACM Symposium on Theory of Computing (STOC), pp. 218–229. ACM Press (1987)
14. Goyal, V., O'Neill, A., Rao, V.: Correlated-Input Secure Hash Functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
15. Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: 14th ACM Conference on Computer and Communications Security (CCS), pp. 466–475. ACM Press (2007)
16. Harnik, D., Ishai, Y., Kushilevitz, E., Nielsen, J.B.: OT-Combiners via Secure Computation. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 393–411. Springer, Heidelberg (2008)
17. Henecka, W., Kögl, S., Sadeghi, A.-R., Schneider, T., Wehrenberg, I.: TASTY: Tool for automating secure two-party computations. In: 17th ACM Conf. on Computer and Communications Security (CCS), pp. 451–462. ACM Press (2010)
18. Hofheinz, D., Unruh, D.: Towards Key-Dependent Message Security in the Standard Model. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 108–126. Springer, Heidelberg (2008)

19. Huang, Y., Evans, D., Katz, J., Malka, L.: Faster secure two-party computation using garbled circuits. In: Proc. 20th USENIX Security Symposium, pp. 539–553. USENIX Association (2011)
20. Huang, Y., Evans, D., Katz, J.: Private set intersection: Are garbled circuits better than custom protocols? In: Network and Distributed System Security Symposium (NDSS) (to appear, 2012)
21. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending Oblivious Transfers Efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003)
22. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding Cryptography on Oblivious Transfer – Efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
23. Jakobsen, T.P., Makkes, M.X., Nielsen, J.D.: Efficient Implementation of the Orlandi Protocol. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 255–272. Springer, Heidelberg (2010)
24. Katz, J., Malka, L.: VMCrypt — modular software architecture for scalable secure computation, `http://eprint.iacr.org/2010/584`
25. Kolesnikov, V., Sadeghi, A.-R., Schneider, T.: Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 1–20. Springer, Heidelberg (2009)
26. Kolesnikov, V., Schneider, T.: Improved Garbled Circuit: Free XOR Gates and Applications. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 486–498. Springer, Heidelberg (2008)
27. Lindell, Y., Pinkas, B.: A proof of security of Yao's protocol for two-party computation. Journal of Cryptology 22(2), 161–188 (2009)
28. Lindell, Y., Pinkas, B., Smart, N.P.: Implementing Two-Party Computation Efficiently with Security Against Malicious Adversaries. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 2–20. Springer, Heidelberg (2008)
29. Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay — a secure two-party computation system. In: Proc. 13th USENIX Security Symposium, pp. 287–302. USENIX Association (2004)
30. Naor, M., Pinkas, B., Sumner, R.: Privacy preserving auctions and mechanism design. In: Proc. 1st ACM Conf. on Electronic Commerce, pp. 129–139. ACM (1999)
31. Nielsen, J.B., Nordholt, P., Orlandi, C., Burra, S.: A new approach to practical active-secure two-party computation, `http://eprint.iacr.org/2011/091`
32. Nielsen, J.B., Orlandi, C.: LEGO for Two-Party Secure Computation. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 368–386. Springer, Heidelberg (2009)
33. Pinkas, B., Schneider, T., Smart, N.P., Williams, S.C.: Secure Two-Party Computation is Practical. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 250–267. Springer, Heidelberg (2009)
34. Shelat, A., Shen, C.-H.: Two-Output Secure Computation with Malicious Adversaries. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 386–405. Springer, Heidelberg (2011)
35. Yao, A.C.-C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (FOCS), pp. 162–167. IEEE (1986)