The impossibility of computationally sound XOR

Dominique Unruh
Saarland University

July 9, 2010

Abstract. We give a simple example that there is no symbolic theory for exclusive or (XOR) that is computationally sound.

1 Introduction

There are two main approaches on how to model and verify the security of protocols. In the computational approach, protocol machines are modeled as computational entities (e.g., Turing machines) that send and receive bitstrings. Cryptographic operations on these bitstrings are modeled as algorithms on bitstrings; the adversary is allowed to perform any polynomial-time computation. In the symbolic approach, messages are not modeled as bitstrings, but as terms over a suitable algebra. The constructors in this algebra model the various available cryptographic operations (such as encryption). The adversary is limited to perform certain well-defined symbolic operations on the terms in his knowledge (e.g., an adversary can derive a plaintext m if and only if he knows the key k and the ciphertext $enc(k, m)$).

Obviously, the cryptographic approach is much closer to reality. The assumption that the adversary will restrict himself to a small set of symbolic operations is not realistic. Yet, there is a big advantage to the symbolic approach. Due to the simple rules that govern the behavior of the adversary, security proofs in a symbolic model tend to be much simpler than in the computational model. In many cases, symbolic security proofs can be found by automated tools while in the computational case, only error-prone, hand-written proofs based on complexity-theoretic reductions are known.

To get the best of both worlds, Abadi and Rogaway [AR02] suggested to study the *computational soundness* of symbolic models. We call a symbolic model computationally sound if the following holds: For any protocol that is secure in the symbolic model, the same protocol is also secure in the computational model. Abadi and Rogaway also gave a first computational soundness result for symbolic models for symmetric encryption; their result was, however, limited to passive adversaries. Subsequent work generalized their approach to deal with active adversaries. There are computational soundness results in the active case for public key encryption (Backes, Pfitzmann, and Waidner [BPW03]; Micciancio and Warinschi [MW04]), for signatures (Backes, Pfitzmann, and Waidner [BPW03]; Cortier and Warinschi [CW05]), for hash functions (in the random oracle model; Cortier, Kremer, Küsters, and Warinschi [CKKW06]), and for zero-knowledge proofs (Backes and Unruh [BU08]).

Yet, an example for a primitive that so-far has defied computational soundness results is the exclusive-or (XOR).[1] Backes and Pfitzmann [BP05] even show that with a common proof technique, no computational soundness result for XOR can be obtained. That proof technique is based on the idea that there needs to be a parsing function that maps bitstrings to corresponding symbolic terms; at the time, it was the only known technique. Since then, however, it was noticed that the parsing function could be implemented lazily in the following sense: If a particular bitstring cannot be parsed, parsing can be delayed until enough information is known (Backes, Hofheinz, and Unruh [BHU09]). The impossibility proof of Backes and Pfitzmann does not apply to proofs based on lazy parsing.

We give a simple example that shows that we cannot expect to get computationally sound symbolic models for XOR. Our example does not only apply to standard symbolic models of XOR (which usually model the commutativity, the associativity, and the cancellation property of XOR), instead, it applies to any "reasonable" symbolic model of XOR. (We discuss what we mean by "reasonable" in the next section.) Our proof is, instead, only based on the assumption that in the computational model, the XOR operation is indeed implemented as a bitwise XOR. We believe this to be a reasonable assumption since otherwise the operation would not be called XOR.

2 The impossibility of computationally sound XOR

In this section, we give an example why computationally sound XOR is impossible. Our example works for essentially all (reasonable) symbolic models. Yet, we do not try to give a precise definition of what a symbolic model is and what the class of reasonable symbolic models is. This is due to the fact that our example is quite simple, and we believe that directly checking whether our example applies to a specific setting would be at least as simple as checking whether an abstract definition of "reasonable symbolic models" is fulfilled.

Informally, we assume the following conditions to be fulfilled by the symbolic model:

- The symbolic model contains a binary operation XOR (written $\oplus$ in the following). We do not impose any conditions on the symbolic modeling of $\oplus$. In particular, we do not assume that $\oplus$ is associative, commutative, or has the cancellation property $x \oplus x = 0$.

- We assume the existence of an infinite set of nonces that can occur within terms. The intuitive meaning of such nonces is that of randomly chosen values. We assume that protocols can use as many different nonces as needed.

[1] The XOR of two bitstrings is assumed to be implemented computationally as the bitwise XOR of these bitstrings. We do not specify what happens if two bitstrings of different length are XORed; this case will not occur in our analysis.

- We assume an equality relation $=$ on terms with the following property: If a nonce C does not occur in a term t, then $t \neq C$.

  This is our strongest assumption, but it is justified by the following observation: Since each nonce symbol N represents a random value, the theory should not make a distinction between the different nonce symbols. In particular, for any terms t_1, t_2, and any permutation σ on the set of all nonces, we expect to have that $t_1 = t_2$ iff $t_1\sigma = t_2\sigma$. Assume now that $t = C$ for some t, C with C not occurring in t. Fix an arbitrary nonce C^* not occurring in t and fix a permutation σ such that $\sigma(C) = C^*$ and $\sigma(N) = N$ for all nonces N occuring in t. Then from $t = C$ we have $t = t\sigma = C\sigma = C^*$. Hence $C = t = C^*$. Thus, all nonces not occurring in t would be considered equal; the set of nonces would effectively collapse to a finite set.

We consider the following (single-party) protocol π:

- First, π sends a fresh nonce C to the adversary.

- Then, whenever π is queried by the adversary,[2] π sends a fresh nonce N_i to the adversary. Let $\mathbf{N}$ denote the set of all nonces N_i sent to the adversary.

- The protocol π expects a list of nonces $M_1, \dots, M_t$ from the adversary.[3] The number t is not fixed; the adversary can send as many nonces M_i as he wishes.

- The protocol π tests whether the following holds: $M_1, \dots, M_t \in \mathbf{N}$ and $M_1 \oplus \dots \oplus M_t = C$.[4] If the test succeeds, π becomes insecure. (E.g., π might leak some secret nonce, raise some bad events, etc., depending on what security means in our particular setting.)

We first claim that the protocol π is secure in reasonable symbolic models. By protocol construction, C is different from all N_i. Thus $M_1, \dots, M_t \in \mathbf{N}$ implies that C does not occur in $M_1 \oplus \dots \oplus M_t$. Hence $M_1 \oplus \dots \oplus M_t \neq C$. Thus the test in the last protocol step never succeeds, and the protocol is secure.

In the computational model, however, π will be insecure. To see this, we need the following lemma:

Lemma 1 *Let $C, N_1, \dots, N_{3n}$ be independently and uniformly chosen bitstrings of length n. Then with overwhelming probability in n, there is a subset $\{M_1, \dots, M_t\}$ of $\{N_1, \dots, N_{3n}\}$ such that $M_1 \oplus \dots \oplus M_t = C$ (where $\oplus$ denotes the bitwise XOR). The set $\{M_1, \dots, M_t\}$ can be efficiently computed given $C, N_1, \dots, N_{3n}$.*

[2] How the adversary queries π depends on the precise protocol model. Typically, the adversary would query π simply by sending a fixed message to π.

[3] If our symbolic model does not permit to encode lists, this can instead be implemented by sending each M_i in a separate message.

[4] If $\oplus$ is not associative, we interpret $M_1 \oplus \dots \oplus M_t$ as $(\dots((M_1 \oplus M_2) \oplus M_3) \dots \oplus M_{t-1}) \oplus M_t$.

Given this lemma, it is straightforward to attack π in the computational setting: Let n be the length of the nonces. Then the adversary request $3n$ nonces $N_1, \dots, N_{3n}$, computes $M_1, \dots, M_t$ as in Lemma 1 and sends them to π. Then the test performed by π succeeds, and π becomes insecure.

Thus, the protocol π is symbolically secure with respect to any reasonable symbolic model, but not computationally secure. Hence, we cannot expect to get a computational soundness result for XOR (unless we restrict the class of allowed protocols such that π is excluded; see also the discussion in the next section).

Proof of Lemma 1. An n-bit string can be seen as an element of the vector space $V := \mathrm{GF}(2)^n$. The addition in this vector space is $\oplus$. Let $S_i := \mathrm{span}\{N_1, \dots, N_i\}$. We first show that with overwhelming probability, $S_{3n} = V$. For this, we use the following notation: We call $i \in \{1, \dots, 3n\}$ good if $N_i \notin S_{i-1}$ or $S_{i-1} = V$. Conditioned on fixed values for $N_1, \dots, N_{i-1}$, we distinguish two cases: If $S_{i-1} = V$, then N_i is good with probability 1. If $S_{i-1} \neq V$, then S_{i-1} is a proper subspace of V, hence $|S_{i-1}| \leq |V|/2$. Since N_i is uniformly chosen from V and independent of $N_1, \dots, N_{i-1}$, we have $N_i \notin S_{i-1}$ with probability at least $\frac{1}{2}$. Let $G_i := 1$ if N_i is good and $G_i := 0$ otherwise. We then have that $\Pr[G_i = 1 | N_1 = n_1, \dots, N_{i-1} = n_{i-1}] \geq \frac{1}{2}$ for all $n_i \in V$ and hence, since G_i only depends on $N_1, \dots, N_{i-1}$, we have $\Pr[G_i = 1 | G_1 = g_i, \dots, G_{i-1} = g_{i-1}] \geq \frac{1}{2}$ for all $g_i \in \{0, 1\}$. Let $X_1, \dots, X_{3n}$ be independently and uniformly chosen from $\{0, 1\}$. Then $\Pr[G_i = 1 | G_1 = g_i, \dots, G_{i-1} = g_{i-1}] \geq \frac{1}{2} = \Pr[X_i = 1 | X_1 = g_i, \dots, X_{i-1} = g_{i-1}]$ for all g_i. (Intuitively, this means that, in the same situation, G_i is at least as likely to equal 1 as X_i is.) Hence, $\Pr[\sum_i G_i \geq n] \geq \Pr[\sum_i X_i \geq n]$. From the Chernoff bound it follows that $\Pr[\sum_i X_i \geq n]$ is overwhelming in n (the expected value of $\sum_i X_i$ is $\frac{3}{2}n$). Hence $\Pr[\sum_i G_i \geq n]$ is overwhelming in n, too. Thus, with overwhelming probability, we have that at least n indices i are good.

Furthermore, we claim that if at least n indices i are good, we have that $S_{3n} = V$. Assume this was not the case, i.e., $S_{3n} \neq V$. Then also $S_{i-1} \subseteq S_{3n} \neq V$. Thus, by definition of "good", for all good i, we have that $N_i \notin S_{i-1}$. Hence $\dim S_i > \dim S_{i-1}$ for each good i. Thus $\dim S_{3n} \geq n = \dim V$ which contradicts the fact that S_{3n} is a proper subspace of V. Thus, if at least n indices i are good, $S_{3n} = V$. Since with overwhelming probability at least n indices i are good, we have that $S_{3n} = V$ with overwhelming probability.

In the case that $\mathrm{span}\{N_1, \dots, N_{3n}\} = S_{3n} = V$, and since $C \in V$, we have that there is, by definition of the span, a linear combination $a_1 N_1 \oplus \cdots \oplus a_{3n} N_{3n} = C$ with $a_i \in \mathrm{GF}(2)$. This linear combination can be efficiently found by Gaussian elimination. Let $M_1, \dots, M_t$ be those N_i with $a_i = 1$. Then $M_1 \oplus \cdots \oplus M_t = a_1 N_1 \oplus \cdots \oplus a_{3n} N_{3n} = C$ and $\{M_1, \dots, M_t\} \subseteq \{N_1, \dots, N_{3n}\}$.

Thus, summarizing, with overwhelming probability we have that $S_{3n} = V$, and in this case, there are $M_1, \dots, M_t$ with $M_1 \oplus \cdots \oplus M_t = C$ and $\{M_1, \dots, M_t\} \subseteq \{N_1, \dots, N_{3n}\}$. These M_i can be efficiently computed. □

3 Discussion

Restricting the protocols. Our counterexample is based on the assumption that we have a protocol model that is powerful enough to express the protocol π described in the preceding section. In particular, this protocol needs to be able to send an unbounded number of nonces, it needs to check whether a given list of nonces is a subset of another list, and it needs to compute the XOR over a list of nonces (this essentially corresponds to a fold operation over a list). Thus, it might be possible that we can derive computational soundness results for XOR if we restrict the class of protocols to, say, protocols that send a constant number of messages and do not contain loops. Yet, models containing only XOR are probably not very useful. If we aim at symbolic models that do not only model XOR but also, say, signatures, we can design the following variant π' of the protocol π. The protocol π' sends only a constant number of messages, and it does not need to keep any state except for a globally shared long-term key pair. We do assume, however, that an arbitrary number of instances of π' can run concurrently.

- We assume a signing key pair shared by all instances of π'. Furthermore, we fix constants c, n.
- When π' receives a message m from the adversary, we distinguish the following cases:
  - $m = \mathsf{c}$. Then π' picks a fresh nonce C and sends (C, σ) where σ is a signature on (c, C).
  - $m = \mathsf{n}$. Then π' picks a fresh nonce N and sends (N, σ) where σ is a signature on (n, N).
  - $m = (\mathsf{n}, m_1, m_2, \sigma_1, \sigma_2)$ where σ_i is a signature on (n, m_i) for $i = 1, 2$. Then π computes $m' := m_1 \oplus m_2$ and sends (m', σ') where σ' is a signature on (n, m').
  - $m = (\mathsf{n}, m_1, \mathsf{c}, c_2, \sigma_1, \sigma_2)$. If $m_1 = c_2$ and σ_1 is a signature on (n, m_1) and σ_2 is a signature on (c, c_2), then π' becomes insecure.

Observe that here, the adversary can obtain signatures on (n, m) where m is an arbitrary term build from $\oplus$ and from nonces N chosen by (different instances of) π'. Furthermore, the adversary can obtain a signature on (c, C) where C is a nonce distinct from the nonces N. In the symbolic setting, C will not occur in any of the terms m such that the adversary knows a signature on (n, m). Hence, the adversary cannot send a message $(\mathsf{n}, m_1, \mathsf{c}, c_2, \sigma_1, \sigma_2)$ that passes the test in the last case of π', so π' is secure in the symbolic setting.

In the computational setting, however, the adversary can get nonces $N_1, \dots, N_{3n}$ with signatures on (n, N_i) from $3n$ instances of π'. Then the adversary uses another instance to get a signature σ_2 on (c, C) for some C. By Lemma 1, the adversary can find a subset $M_1, \dots, M_t$ of the nonces N_i such that $M_1 \oplus \cdots \oplus M_t = C$. Then the adversary uses $t-1$ further instances of π' to obtain signatures on (n, m_i) with $m_i = M_1 \oplus \cdots \oplus M_i$ for $i = 2, \dots, t$. Note that $m_t = C$. Thus the adversary now has a signature σ_1 on (n, C). Finally, the adversary sends $(\mathsf{n}, C, \mathsf{c}, C, \sigma_1, \sigma_2)$ and the protocol becomes insecure.

We stress that the counterexample π' still does not exclude that there could be small classes of protocols for which computational soundness results for XOR might be possible. We believe, however, that in order not to contain π', that class would have to be very restrictive indeed.

Passive adversaries. Our counterexample is based on the assumption that the adversary is active. In fact, in the passive setting, computational soundness results for XOR are known (Kremer and Mazaré [KM07]).

On dynamic symbolic models. We have claimed that all reasonable symbolic models have the property that $t \neq C$ whenever the nonce C does not occur in t. We have based this claim on the fact that due to symmetry reasons, $t = C$ would imply $t = C'$ and hence $C = C'$ for all C'. One possible way out of this might be to use "dynamic symbolic models". The idea would be that the equality $=$ is not fully defined from the very start. Instead, the adversary may, during runtime, create new equality rules $t = u$. For example, after seeing $N_1, \ldots, N_s, C$, the adversary could pick a subset $M_1, \ldots, M_t$ of $N_1, \ldots, N_s$ and add the equality rule $M_1 \oplus \cdots \oplus M_t = C$. Of course, the rules that the adversary can add must be subject to suitable restrictions. For example, it must be ensured that the adversary can never add rules that make all nonces equal, or rules that make a nonce known to the adversary equal to a key that is supposed to be unknown to the adversary. We do not know whether this approach is viable, and it seems that it would lead to a very complex symbolic model.

Other computational implementations. We have assumed that $\oplus$ is indeed implemented as a bitwise XOR. Furthermore, we have assumed that nonces are uniformly random bitstrings (in particular, they contain no type-tagging or similar headers). We believe that the second assumption can be relaxed; for this, Lemma 1 needs to be extended to the case where the nonces are chosen with respect to a particular nonuniform distribution. It is an interesting question whether Lemma 1 actually holds for any distribution of nonces. The assumption that $\oplus$ is implemented as a bitwise XOR seems necessary for our proof. If, e.g., $\oplus$ would be the multiplication in some cyclic group in which each element is self-inverse, it might be computationally infeasible to find M_i such that $M_1 \oplus \cdots \oplus M_t = C$. Although such an operation $\oplus$ would not merit the name "XOR" any more, such an operation might still be useful as a computationally sound replacement for XOR in existing protocols.

Other cryptographic primitives. Although our counterexample only touches the computational soundness of XOR, it may serve as a warning. There are many cryptographic primitives that have a strong algebraic structure, e.g., homomorphic encryption or many blind signature schemes. A symbolic analysis of protocols based on such primitives may be subject to similar problems as the protocol π. Thus, before trusting a symbolic security analysis for protocols based on such primitives, we strongly advocate to study the computational soundness of these primitives.

References

[AR02] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, 2002.

[BHU09] Michael Backes, Dennis Hofheinz, and Dominique Unruh. CoSP: A general framework for computational soundness proofs. IACR ePrint 2009/080, February 2009. *Only the version from February 2009 discusses lazy parsing.*

[BP05] Michael Backes and Birgit Pfitzmann. Limits of the cryptographic realization of Dolev-Yao-style XOR. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *Lecture Notes in Computer Science*, pages 178–196. Springer-Verlag, 2005.

[BPW03] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A composable cryptographic library with nested operations (extended abstract). In *Proc. 10th ACM Conference on Computer and Communications Security*, pages 220–230, 2003. Full version in IACR Cryptology ePrint Archive 2003/015, Jan. 2003.

[BU08] Michael Backes and Dominique Unruh. Computational soundness of symbolic zero-knowledge proofs against active attackers. In *21st IEEE Computer Security Foundations Symposium, CSF 2008*, pages 255–269, June 2008. Preprint on IACR ePrint 2008/152. To appear in the Journal of Computer Security.

[CKKW06] Véronique Cortier, Steve Kremer, Ralf Küsters, and Bogdan Warinschi. Computationally Sound Symbolic Secrecy in the Presence of Hash Functions. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2006)*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2006.

[CW05] Vèronique Cortier and Bogdan Warinschi. Computationally sound, automated proofs for security protocols. In *Proc. 14th European Symposium on Programming (ESOP)*, pages 157–171, 2005.

[KM07] S. Kremer and L. Mazaré. Adaptive soundness of static equivalence. In *Proc. 12th European Symposium on Research in Computer Security (ESORICS)*, Lecture Notes in Computer Science, pages 610–625. Springer-Verlag, 2007.

[MW04] Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proc. 1st Theory of Cryptography Conference (TCC)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.