Sketchable Security

Dan Brown[*]

April 1, 2007

Abstract

Sketchable security is shown to be a viable successor to provable security. Whereas security proofs can be found to have flaws, gaps, or to be just plain hard to interpret let alone understand, security sketches escape these problems. Moreover, security sketches can be made more plausible and understandable to a wider audience. This document describes some of the most salient points of sketchable security. It also systematizes a strategy for successful sketchably secure systems. A theorem, whose proof is sketched, is given about the seven stages of sketchable security.

1 Introduction

Provable security has gradually risen in popularity among theoretical cryptologists over the years. While provable security obviously provides many tremendous benefits, it has certain pitfalls. Security proofs can get very complicated and difficult to verify. Because of their complexity, there can be gaps and flaws that are discovered later. Very few readers of complex security proofs are likely to accurately appreciate their benefits, let alone verify their accuracy. Also, there sometimes seems to be some kind of barrier to provable security, where a scheme that seems secure seems impossible to prove to secure, or is actually impossible to prove secure.

This paper presents an alternative to provable security: sketchable security. By sacrificing some of the rigor of provable security, sketchable security can overcome some of the difficulties and limitations of provable security.

The hope of sketchable security is to retain as many of the benefits of provable security as possible, while overcoming as many of the deficiencies as possible. Furthermore, for some particular seemingly-secure schemes, it may prove too difficult — or even provably impossible — to prove them secure. In these cases, a security sketch may be the only possible argument.

1.1 Related Work

It could be argued that sketchable security already exists. A cynic may argue that all papers in provable security are actually just sketchable security, given that some proofs of $1 + 1 = 2$ have consumed over a hundred pages, and most papers in provable security are (a) less than 100 pages, (b) use the number 2 or hypothesized larger numbers, and (c) don't cite the papers proving that $1+1 = 2$. However, in this paper, we will take the view that such lack of rigor really reflects a general consensus among mathematicians and computer scientists of a commonly accepted groundwork of

[*] Author of *The Da Vinci Cipher* (to appear)

basic facts not requiring citation. Accordingly, any serious attempt at actually proving security, provided it is not by some lunatic, belongs to the field of provable security, not sketchable security. This is not to say that all attempts at proofs are correct, just that to qualify as a security sketch, the authors of the paper must not be asserting, implicitly or explicitly, that the argument is completely rigorous.

Note that calling an argument a proof is generally an immediate disqualifier for a security sketch, unless it is balanced very heavily with qualifications that the proof is actually only a sketch, and it is not really rigorous. Conversely, it is very customary for provers to qualify the proofs as sketches or outlines. Such terms, under our rather stricter definitions, do not alone qualify the argument to be security sketch, because in these cases it is very frequently the case that the argument is only a sketch or outline in the sense that the author is intentionally omitting some pesky details which will be written up in a "full" version of the paper. Arguably, if the "full" version never materializes, then we can retroactively designate the claimed sketch as just that: a sketch. Obviously, if an author is using "sketch" just to be modest or cautious, when upon further inspection the argument is perfectly rigorous, or really just an attempt to be so, then the work belongs to the field of provable security, not sketchable security.

A less common entity in cryptology research is a heuristic argument (note that it is somewhat more common in essays about mathematical conjectures). We will not take the blanket view that all such heuristic arguments of security are security sketches. This may seem paradoxical at first, because the term heuristic generally means non-rigorous. Our justification, however, is that most heuristic arguments are generally quite rigorous and logical, but usually involve some large assumptions that the author has no idea how to prove. In conventional mathematics, a heuristic argument could reasonably be designated a sketch, but in cryptology (and computer science to an extent), unproven assumptions are simply par for the course. Therefore, a "heuristic argument" in cryptology is usually just what a rather shy author would call what a bold author would call a security proof (under some unproven assumptions).

Some heuristic arguments, one may contend though, make false assumptions. Although this is an extremely fascinating idea, one may note that it is not customary to call such a thing a "sketch". One of our goals of sketchable security is sacrifice some rigor for greater plausibility and understandability to a wider audience. Making a false assumption, in the humble view that we take in this paper, decreases both plausibility and understandability, at least to a less sophisticated audience. Accordingly, we leave rigorous arguments based on false assumptions as outside the scope of sketchable security.

It is hoped that the comparison above between provable security and sketchable security should suffice to make clear the distinction between the two distinct fields. We also need to distinguish sketchable security from the other end of spectrum of cryptology, which for lack of a better term, one might call practical cryptology (as opposed to theoretical cryptology).

Papers in practical cryptology may have appeared containing casual comments, with no attempt at a proof, that particular schemes are secure. In light of the discussion above, one is likely think that these are prime examples of security sketches. One must be cautious here too. Firstly, these papers are usually proposals of new schemes. One must distinguish informal assertions and conjectures of security from security sketches. An author may state that some particular line of attack fails, and therefore that the scheme may be secure (implicitly implying that there are probably no other attack strategies). In this paper, we will not call this a security sketch. We require a security sketch to be more formal. It must not be presented casually as some informal side remarks: it must be put boldly front and center for the reader to inspect and chew over. Furthermore, a security sketch is meant to be a substitute for a security proof, so the gist of the security sketch must be that any attack is impossible, not merely a weaker assertion that author thinks that all attack strategies the author can think of fail. As such sketchable security is higher hurdle than such casual comments, or even the frequently appearing formal reports of failed cryptanalysis.

1.2 Organization of this Paper

Seven sections or subsections, excluding this one §§1.2.

2 The S^7 System

In this paper, not only do we discuss sketchable security, but we describe to you a strategical system for using and applying sketchable security, which we designate S^7, which stands for the objective "Sketchable Security Should Silence pSeudoScientific SkepticS" and also for its seven stages. Of course, sketchable security and the S^7 system represent both a paradigm shift and shifty proposition.

Under the S^7 system, cryptographic schemes sketchably secure in the standard model can slide safely (not just sneak) into serious standards, be they state (NIST), society (IEEE), or socialist (IETF) standards.

2.1 Devising a Security Sketch

Suppose that you have a cryptographic system, which could be a pre-existing scheme, or preferably one of your own making. Devise a strategy to come up with a plausible security proof. Sketch this out, being sure to omit or overlook some crucial details. If you inadvertently come up with a completely rigorous proof, then you should probably abandon the scheme and try another scheme, because, as we've stated earlier, we do not consider to-be-completed proof to be a proper sketch, unless it actually cannot be completed. Of course, if you do find a security proof, you definitely have the option, if it's your thing, to write a provable security paper on the scheme that you proved secure. This paper has, of course, has no further advice for you on that task.

When writing your security sketch, use highly informal notation if your reader will decipher your intentions, or highly formal notation if your reader prefers deciphering twenty variables in ten fonts. Optionally, use awkward language if you wish the reader to focus on mathematical-like parts of the sketch, or use eloquent language if you prefer to edify the reader. In either case, it's good to slightly dazzle the reader, because impressing the reader enhances plausibility (though be careful not to undermine understandability). Evidently, with respect to these two parameters (notation and language) of cryptology research, the S^7 system of sketchable security is highly flexible, making it suitable for all sorts of researchers and readers. In summary, the appearance, in terms of the level of formality, of security sketch should be similar to a security proof, without having all the mucky rigor for the reader to get distracted by.

Simply write "Sketch:" before your security sketch. As in provable security, this should be preceded by a statement of what you wish to sketch (the statement could be labeled "Theorem" or "Fact", but do not use "Proposition", since it sounds too iffy). Alternatively, put "Proof (Sketch):" before arguments, as this often appears in provable security and other areas of mathematics. More subtly, put "Proof:", but make it explicit or implicit, inside or outside the "proof" that it is all really just a sketch. It mainly depends on your confidence level, or the impression that you wish to convey thereof. The label decision is a secondary aspect of S^7 system, but the sketch must be clearly labeled somehow, to avoid deteriorating into an unconvincing casual comment. That said, you are free to surround the security sketches with plenty of sketchy casual comments.

The crucial aspect of sketchable security is to sacrifice rigor for greater understandability, and plausibility to an audience at large, extending beyond the narrower audience of theoretical cryptologists.

2.2 Seven Stages of Sketchable Security

The S^7 is not limited to devising security sketches. Once you have such a paper, the S^7 system gives you a strategy to apply the security sketch to the real world. The system works in seven stages, which must each be pursued systematically. These stages bear some resemblance to life cycle for a provable security result, but note that to the best of my knowledge, they have not been formalized for provable security. With sketchable security and the S^7 system, the seven stages are an integral part of the system.

1. First is rejection (preprint optional).
2. Second is acceptance (to some conference).
3. Third is bandwagon-hopping (where others steal your sketch technique for their own schemes).
4. Fourth is standardization (modification mandatory, deployment optional).
5. Fifth is somebody[1] finding a flaw in the sketch (falsifying the proof).
6. Sixth is somebody[2] fixing the flaw in the sketch (re-truthifying an amended proof).
7. Seventh is somebody[3] finding a serious attack on the cryptographic scheme.

The beauty of the seven stages is the astonishing S^7 principle that each stage takes seven times as long as the previous. This is only an asymptotic, of course. Even so, it is evidentially exponential. In particular, we have the spectacular special case:

Theorem 1. *If a flaw in a security sketch for a scheme S is found only after seven years, then an attack on the scheme S will only be found after 7^7 years.*

Proof (Sketch). Suppose that once the security sketch was accepted for publication (second stage), it took cryptologists seven years to find a flaw (fifth stage).

If it takes them that long to merely find a flaw in the sketch, it will take them exponentially longer to find an attack, which is 7^7 years. □

Of course, this is just an upper cryptanalytic (or lower cryptographic) bound, but because it is only asymptotic, it should silence any skeptics of your system. Provided you get through acceptance and bandwagon-hopping, you may proceed to standardization.[4]

Since two sketches are as good as one (unless they contradict), we provide a second sketch: put $S = 7$, and get 7^7. This depends on the rather technical S^7 lemma, which will be detailed in an appendix of a conference version (either CRYPTO or EUROCRYPT) of this journal article. The S^7

[1] Best if it's you.

[2] Best if it's you.

[3] Best be gone.

[4] Do not collect \$7.

lemma is too detailed and unimportant to state here, though the apt reader should have surmised its statement, if not its sketch, already. The point is that the two sketches make Theorem 1 *doubly true*. By the contrapositive, it isn't clearly not a *double negative*, which should be re-assuring.

When deploying the S^7 strategy, be sure to prolong the early stages, since that should delay the discovery of an attack (seventh stage). Do not overly delay, however, so that you can get the benefit of some of the intermediate stages. A good delay should also give you enough time (by Theorem 1) or excuse ("it was just a sketch") to escape any blame and shame from the first, fifth or final seventh stages.

3 Sketchably Secure Encryption

By way of an example, we now illustrate an application of the part of the S^7 system for devising a security sketch. Suppose that we want to encrypt a message. We do so as:

Theorem 2. *Let M be a message. Let H be a secure hash function. Then $H(M)$ is a secure encryption of M.*

Sketch. Recall that a hash means a mess or a jumble, so to hash a message we just encrypt it. Therefore a hash of message is an encryption of a message.

If the hash is secure, then the encryption is secure. Conversely if the encryption is secure, then so is the hash. Indeed, if E is a secure encryption, and we define $H(M) = E(M)$, then an attacker cannot find M from $H(M) = E(M)$, because the encryption E is secure. Therefore H is one-way. The attacker cannot find two distinct messages M and M' such that $H(M) = H(M')$, because then $E(M) = E(M')$ which implies that $M = M'$ because we can just decrypt the encryptions. Therefore H is collision-resistant. Because H is one-way and collision-resistant, it a secure hash. Therefore, a secure encryption = secure hash, so secure hash = secure encryption. □

In particular, we have as a useful corollary that $SHA-256(M)$ is a secure encryption of message M, at least until a collision is found $SHA-256$, in which case one can just apply hash-encryption, say $SHA-512(SHA-256(M))$.

As a second, more efficient, example

Theorem 3. *Let M be a message. Then $E(M) = M \oplus M$ is a secure encryption scheme, providing you only send each message once.*

Proof (Sketch): Shannon's theorem for one-time pads is that if K is secret and only used once, then the scheme $E(M) = K \oplus M$ is a secure encryption scheme. Put $K = M$. If you only encrypt M once, then K is only used once and you have a one-time pad. Of course M is secret, otherwise you wouldn't need to encrypt it. Therefore the encryption is secure. □

While more efficient, being a simple stream cipher, this example has a longer ciphertext than the hash-based encryption.

The two examples are just illustrations. With a little more work, you may find sketchably secure encryption which is much more efficient or shorter, or that has some extra useful properties, such as message recovery. We give just one more example, providing message recovery, perhaps artificially complicated.

Theorem 4. *Let M be a message. Divide M into four parts of equal length as $M = M_0 \| M_1 \| M_2 \| M_3$. Define $E(M) = M_0 \oplus M_1 \| M_1 \oplus M_2 \| M_2 \oplus M_3 \| M_0 \oplus M_2 \oplus M_3$, provided that message M is only encrypted once and has distinct parts.*

Proof. The sketch of security is similar to the previous sketch. Define keys $K_0 = M_1$ and $K_1 = M_2$ and $K_2 = M_3$ and $K_3 = M_0 \oplus M_2$. Note that each key only used once, because M is only used once and the parts of M are distinct. Now the encryption of M is $K_0 \oplus M_0 \| K_1 \oplus M_1 \| K_2 \oplus M_2 \| K_3 \oplus M_3$, each part being a secure one-time pad with a secret key used only once. Since each part is secure, the whole is. □

The only remarkable aspect of this scheme is the message recovery. It is somewhat surprising that a secure encryption scheme can simultaneously provide message recovery, since these appear contradictory properties. Suppose the ciphertext is $E(M) = C_0 \| C_1 \| C_2 \| C_3$, when divided into four equally long parts. We can decrypt the ciphertext, assuming that we know the key E, as follows $M = C_2 \oplus C_3 \| C_0 \oplus C_2 \oplus C_3 \| C_0 \oplus C_1 \oplus C_2 \oplus C_3 \| C_0 \oplus C_1 \oplus C_3$. Evidently, decryption is a lot more work than encryption, involving around twice as many operations, so an attacker will be unable to decrypt. More precisely, if the encryptor does work 2^{40} bit operations to encrypt a message, then the scheme has 80 bits of security. We leave the sketch of this exact security analysis as an exercise for you, the reader.

4 Conclusion

Use the S^7 system for sketchable security, and you too will one day see all the same successes as the most successful cryptosystems in both theoretical and practical cryptology. Although the S^7 system is a matter of art, done well, it is as effective as any science.