Anomalous Look at Provable Security

Douglas Wikström

KTH Royal Institute of Technology
dog@kth.se

February 7, 2019

Abstract. We observe that if a party breaks one cryptographic assumption, construction, or system, then it can reduce the trust in any other. This highlights a shortcoming in the common interpretation of the provable security paradigm that may lead to unwarranted trust. This may have practical implications.

Then we argue that the provable security paradigm remains sound in applications provided that assumptions are made with care. We also strengthen the argument for the study of combiners and constructions based on generic assumptions, and transparent standardization processes in applied cryptography.

1 Introduction

This note is motivated by a real-world need to interpret, explain, and defend the provable security paradigm to a broader audience of scrutinizing scientists and philosophers. The author implemented a provably secure mix-net [12] based on the discrete logarithm assumption which has been used by local and national election authorities in Israel, Spain, Norway, and Estonia. The reader should have no trouble imagining that the stakes are high, that the adversary may be a nation state actor, and that trust plays an unusually strong and long-lived role in this application; a perfect target for the smear campaign described below.

Most modern cryptography rely on computational assumptions. Consider an NP relation R and the corresponding language L_{R}. The simplest form of problem P used in cryptographic constructions consists of computing a witness w for an instance x such that $(x, w) \in \mathsf{R}$ where x is drawn from $\mathsf{L}_{\mathsf{R}} \cap \{0, 1\}^n$ according to some distribution. An assumption A about P states that no algorithm can solve P using less than $T(n)$ time or $S(n)$ space (or some other resource), except with negligible probability. The size of the instance drawn randomly is determined by the security parameter n. More generally, we can consider interactive games or make assumptions about how hard it is to achieve a given goal. The type of problem considered only changes the details of the argument put forth here.

Cryptographers often construct and analyze the security of primitives and protocols under computational assumptions. A definition of security is in fact itself a definition of a problem parametrized by concrete constructions. For example, the definition of security for signature schemes [4] is based on the following experiment: (a) the adversary is given a public key, (b) it is allowed to ask for any number of signatures of messages of its choice, and (c) it outputs a message and a candidate signature. The adversary is successful if it has not previously asked for a signature of the message it outputs and the candidate signature is valid. This becomes an interactive computational problem for any concrete signature scheme; to be solved by the adversary.

At a high level a proof of security is therefore a reduction of one problem P to another problem P' and a complexity analysis showing that the reduction is efficient. For the purpose of this note we simplify and consider an assumption about a problem to be either:

1. true in the sense that the time, space, or whatever resource is required to solve a problem is exponential in n, or
2. false in the sense that there is a practical method for solving the problem in polynomial time in n.

In reality there are of course problems that are neither very hard nor very easy to solve and there are problems that can be solved with some probability of success.

2 Why Do We Need Multiple Assumptions?

Cryptographers have spent considerable effort identifying the key properties needed to construct a secure instantiation of any given notion. The most fundamental assumption is perhaps that one-way functions exist, from which one can construct and prove the security of: pseudo-random generators, pseudo-random functions, signature schemes, etc (see [3] for references), but there are not many assumptions that are efficient enough to be used in real-world applications, in particular for public key cryptography. Problems underlying assumptions include: factoring integers, computing discrete logarithms, finding a shortest vector in lattices, and learning with errors.

Each assumption comes with pros and cons for specific cryptographic tasks, so they are not equivalent from a constructive perspective. All assumptions are of course not equally trusted either. The most trusted assumptions used in the cryptographic literature are often called *standard*, whereas those that have not yet earned the trust of most cryptographers are called *non-standard*, despite that there is no strict consensus in the community that separates them.

Thus, there are several reasons why it is important to have multiple assumptions to choose from when designing primitives and protocols, but arguably the most fundamental is to have alternatives if one assumption turns out to be false.

However, this line of thinking is based on the belief that whoever discovers an algorithm that solves a problem, and thereby breaks an assumption, shares this knowledge (or that the announcement of a solution makes it significantly easier for others to find it). Below we argue that there is reason to doubt this belief and that this should influence how we do research in cryptography.

3 Beliefs About Assumptions

There either exists an algorithm that solves a problem efficiently or not (assuming our simplified dichotomy above). Given that an algorithm exists one may ask if it has been *discovered*, and if it has been discovered if it is *publicly known*. There are shades of gray, but for the purposes of this note we can quantify our belief in a given assumption A_i with respect to knowledge and time by encoding these alternatives as binary random variables.

We let D_i indicate that an algorithm that breaks A_i is discovered and announced convincingly, but kept private. We let K_i indicate that an algorithm that breaks A_i is discovered and made publicly known. We may now for a given time period, say Y years, assign probabilities based on our belief, i.e., we may set $\Pr[D_i = 1] = \delta_i$ if we believe that the probability that an algorithm for solving A_i is discovered at all within the time period is bounded by δ_i. For any standard assumptions and a reasonable time period the probability δ_i is small.

Note that this belief takes on a concrete meaning for data or resources with value v that are protected by cryptographic notions that rely on the assumption, since the expected loss of value is then bounded by $\delta_i v$.

One would expect that if it is known that an algorithm exists, but is not yet made public, then it would be discovered more quickly, since more researchers would study the assumption. This is encoded in the conditional probability, i.e., we expect that $\Pr[K_i = 1 \mid D_i = 1] = \omega_i$ for a somewhat larger ω_i, but given that the number of assumptions is small, δ_i is small, and the academic reward in finding an attack is large, we would expect ω_i to be within a small constant factor of δ_i.

This is of course a simplistic model of what in reality is a continuous time random process, but it suffices to present our argument.

4 A Smear Campaign

Denote by A_i an assumption about a problem P_i for $i \in I$, in some index set I, and suppose that Jenny the genius has solved P_j for some j and that she convincingly announces the discovery without disclosing anything except that $j \in B$ for some subset $B \subset I$. In other words the world knows that $D > 0$, where $D = \sum_{i \in B} D_i$, but nothing else. We call this the *smear campaign* and consider some implications below. We choose to state the observations below informally, since the calculations are straightforward and the philosophical implications are more interesting. The reader is invited to consider numerical examples and examine the history of cryptanalysis to derive quantities.

Suppose that we have protected data and resources with value v_i that rely on cryptographic constructions and protocols based on A_i. Without Jenny's announcement our belief is that the loss during the foreseeable future is $C_i = D_i v_i$. This is acceptable when δ_i is small, since given our beliefs the expected loss is $\mathrm{E}[D_i v_i] = v_i \delta_i$. The question is what changes when Jenny convincingly announces that $D > 0$?

We have $\mathrm{E}\left[C_i \,|D > 0\right] \geq v_i\delta_i/\sum_{\ell\in B}\delta_\ell$, since based on our belief we can bound $\Pr\left[D_i = 1\,|D = 1\right]$ from below by $\delta_i/\sum_{\ell\in B}\delta_\ell$. If $k = |B|$ is relatively small and all δ_i are within a factor h, i.e., $h = \max_{j,\ell\in B}\{\delta_j/\delta_\ell\}$, then we may conclude that $\mathrm{E}\left[C_i \,|D > 0\right] \geq v_i/(hk)$ for every $i \in B$. This means that Jenny must break an assumption that is reasonably trusted, or else h would be very large, but there are several assumptions to choose from (and some may even be manufactured by Jenny to contain a backdoor and only appear to be hard to break to us).

Define $\mathsf{P}_{i,h}$ to be the problem of solving any problem P_j such that $\delta_j \leq \delta_i h$. If Jenny solves $\mathsf{P}_{i,h}$, then she can set $B = \{i, j\}$ to smear dirt on the assumption A_i, or include other indices in B to smear dirt on multiple assumptions, albeit more thinly. This may be valuable to Jenny, since finding a solution to a suitable problem P_j is still costly. The real probability $\alpha_{i,h}$ that Jenny solves $\mathsf{P}_{i,h}$ may be significantly larger than δ_i due to the freedom to choose multiple problems to attack, so our best conservative guess $\delta_{i,h}$ of the value of $\alpha_{i,h}$ should also be larger than δ_i.

Example 1 (Signature schemes). Suppose that Alice is given a signature of a financial commitment made by Bob and that the signature scheme is provably secure under the shortest vector problem in a lattice. If Jenny discovers an efficient factoring algorithm (or builds a quantum computer) and shows that she can either break the shortest vector problem or factor large composite integers, then Bob's signature can no longer be fully trusted since to an outside observer the shortest vector problem may well have been solved.

Although the observation is straightforward, the example show that the problem of trust is largely hidden in how proofs of security are *interpreted*.

Implementing the smear campaign is easy. For typical computational assumptions Jenny can design a zero knowledge proof of knowledge that takes $x = (x_i)_{i\in B}$ as input and proves knowledge of w such that $(x_i, w) \in \mathsf{R}_i$ for some $i \in B$. Then she can derive the instance x as the output of some unpredictable public process, using a hash function as a random oracle, etc, and use the protocol she designed to show that she can break A_i for *at least one* $i \in B$. We stress that Jenny can do this anonymously with virtually zero risk of detection in practice.

5 Interpreting Proofs of Security

The observation may seem counter-intuitive in that the insecurity of one construction may influence the trust in another, but it is quite natural when considered through a Bayesian lense.

Recall that we may view a definition of security for a cryptographic notion as a problem $\mathsf{P}[x]$ (parametrized by a construction x) to be solved by the adversary and that any construction c gives a problem $\mathsf{P}[c]$. We say that c is provably secure relative the definition of security $\mathsf{P}[x]$ under the assumption A if the existence of an efficient algorithm for solving $\mathsf{P}[c]$ implies the existence of an algorithm for solving the computational problem P in a way that contradicts A.

The problem is that this type of model does not contain any explicit representation of trust. A commonly made argument, e.g., implicit in key size recommendations [2], proceeds as follows.

1. Use what is known about the manufacturing of computers to predict that no computer will be able to execute T steps in the next Y years.
2. Use what is known from the attempts to solve the problem P to argue for a belief that the probability that an algorithm is found within the next Y years that solves P in time T is smaller than δ.
3. Prove the security of construction c, i.e., describe and analyze a reduction that shows that if $\mathsf{P}[c]$ is solved in time T', then P can be solved in time T.
4. Conclude that if our belief about P is correct, then with probability δ the problem $\mathsf{P}[c]$ cannot be solved in time T' within the next Y years, i.e., c is secure relative the definition $\mathsf{P}[x]$ and time T'.

Although the logic of this argument is sound, it is still flawed as a security argument. The problem is that $\mathsf{P}[x]$ typically captures the *actual* security goals G implicitly and they rely on more than the hardness of P; other parties must *believe* that P is hard. More precisely, there is an implicit additional step in the argument that we have never seen made formally.

5. If with probability δ the problem $\mathsf{P}[c]$ cannot be solved in time T' within the next Y years, then with probability δ the security goals G are satisfied during the next Y years.

For many notions this deduction is invalid, since it is missing the premise that other parties may have to *trust* that P is hard. The smear campaign shows explicitly that it suffices to break any assumption to damage this trust.

Thus, although c satisfies its security definition $\mathsf{P}[x]$ under the assumption that P is hard, and the actual security goals G also follow logically as long as all relevant parties believe that P is hard, it does *not* follow that c guarantee the security goals G without such a belief even if P is truly hard. The problem can be illustrated by considering signature schemes.

Example 2 (CMA Security vs. Non-repudiation of Signatures). A signature scheme that is provably CMA secure under assumption A if nobody can forge a signature even when given access to a signature oracle, but the implicit actual security goal is non-repudiation, i.e., that nobody will believe a party that refuses to acknowledge that it produced a signature.

The problem is we require both: (1) that the signature scheme is provably secure under A, and (2) that others *believe* that A is true, to conclude that non-repudiation is guaranteed. The latter may be false even if A is logically true.

The reader should object at this point. It is well known that digital signatures are different from an analog written signatures in that the signer, e.g., can claim to have lost the secret key, and any outside observer must *believe* that this was not the case to trust any claim made by the holder of a signature. Other external events may also influence beliefs. How is the smear campaign different?

It is not different in principle, but the difference in degree is striking. No cryptographer in the world can, e.g., claim that any cryptographic construction that is provably secure under the RSA assumption is secure if there is overwhelmingly strong evidence that either the RSA problem or the shortest vector problem has been solved (and no additional information is available). Claiming that this is unlikely to happen is merely a different way of saying that we are willing to believe that all standard assumptions are true, which is what the smear campaign allows an adversary to force us to do (at least for some applications).

It is important to keep in mind that the problem of lack of information and hidden variables is shared with all natural sciences and is summarized in the Duhem-Quine thesis: we cannot test a hypothesis in isolation due to dependencies with variables outside the experiment. What makes the situation particularly difficult to manage in cryptography is that: the environment is adversarial, experiments are hypothetical, and the assumptions are poorly supported by evidence (compared to natural sciences).

6 This is Not a Criticism of Provable Security

The naive interpretation is that the provable security paradigm is flawed and that many definitions of cryptographic notions should be augmented with random variables that represent trust, since otherwise provable security does not necessarily imply security as intended even for basic notions like signature schemes.

We do think that our observations say something interesting, but we do not agree with the naive interpretation. Given the discussion about provable security sparked by Koblitz and Menezes [8] it is perhaps worthwhile to state this explicitly. A hyperrealistic approach that includes a mathematical representation of trust would indeed give a more faithful model of reality, but there are several reasons to not choose this approach.

Cryptography is a branch of mathematics and there is no inherent reason why any results should be realistic (whatever that means) if they provide a deeper understanding of the interplay between computation, interaction, and knowledge. A hyperrealistic approach would likely clutter theory with inconsequential details. For example, even if our example shows that CMA security does not imply non-repudiation in the sense it is often understood, it does capture the essence of the problem of forgery.

In applied cryptography the model of computation and communication must be more realistic to allow drawing useful conclusions about the real world, e.g., reductions should be exact instead of asymptotic. This already makes the theory hard to work with and apply. We think this is a real problem that must be addressed, but this is not a criticism of cryptography no more than pointing out that numerical instability is a serious issue in many applications of analysis.

Introducing a hyperrealistic model would, at least in the foreseable future, make matters worse for practitioners and not give a more precise understanding of real security, since the uncertainty about the assumptions is large to start with and there are many other unrealistic aspects of cryptographic models that may be at least as important to model with more precision.

7 What About Combiners?

The most common type of combiner (see [6] for a first systematic treatment) takes k concrete instantiations of a cryptographic notion and constructs another that remains secure provided that at least t of the original instantiations are secure. Even if we learn from Jenny that exactly one assumption is broken, we may at best claim that the probability that our combiner is broken during the foreseeable future is bounded by something like $\max_{S\subset[k],|S|=t-1}\{\prod_{i\in S}\delta_i\}$ provided that the D_i's are independent.

Unfortunately, assumptions are rarely independent in an empirical sense unless they originate from very different sources, e.g., the factoring and discrete logarithms problems are not "independent", but they may be fairly "independent" of shortest vector problems in lattices although in principle many problems can be encoded in lattice form.

Jenny may cast doubt over a combiner without breaking any of its underlying assumptions, but it is less convincing and/or leaks information. Consider a combiner based on assumptions A_1 and A_2 and assume that Jenny breaks assumption A_3 and proves that she broke A_1 or A_3 as well as A_2 or A_3. This lowers trust in the combiner, but it also increases the probability that A_3 is the culprit. To avoid this, Jenny would have to involve yet another assumption that she broke.

Thus, on the one hand using combiners is an important tool to counter the smear campaign and we think we should focus more on discovering, standardizing, and using combiners, but on the other hand combiners do not necessarily provide as strong protection against the smear campaign as one may expect.

8 Why Not Exploit an Attack Directly?

One may argue that being able to break cryptographic schemes is more valuable than what Jenny has in mind, since it, e.g., gives access to secret information, but this is not necessarily true. The value of information is highly dependent on who gains access to it and how it can be used strategically or be monetized.

It may be easier to monetize the damage done to society at large with low risk of being identified using the smear campaign than to try to sell the attack or use it and risk detection due to actions correlated with the information. A terrorist organization, or nation state actor, may even have as primary goal to cause harm to a world that relies on cryptographic algorithms and protocols. Furthermore, Jenny may still be able to exploit the discovery or attack at a later date even if she first uses the smear campaign as the first step of a grander malicious strategy.

9 Variations and Applications

Although we present the observation in terms of computational assumptions in this note for clarity, the same principle applies in other security-related settings where it seems harder to mitigate. Below we give some examples.

Recall that we think of computational problems in a broad sense, so breaking any given construction is considered solving a problem. There is some suspicion that NSA embedded a backdoor in a pseudo-random generator [10]. Less concrete concerns have been voiced about light-weight symmetric primitives originating from NIST that were proposed as a standard to ISO. A common argument against embedding trapdoors in standardized constructions is that the risk that the trapdoor is found and security or trust is damaged is too high. The smear campaign strengthens this argument in that Jenny can cast doubt on all standardized constructions and effectively force the standardization organization to either: (a) point out the construction that contains the trapdoor and admit knowledge of it, or (b) accept decreased trust in all its standardized constructions.

Another example would be an attack on one of the many different crypto currencies. The vulnerabilities in IOTA [7] (in the form of a hash function that was not collision resistant) and the flaw in Zcash [11] (allowing counterfeit coining), show that this is not far fetched. If Jenny found such a vulnerability, then she could have deliberately caused notable volatility on the market for *all* crypto currencies and traded on the volatility using financial derivative instruments.

Another real-world opportunity for Jenny was the recent weakness discovered in Infineon's software library used to generate keys for the Estonian smartcard public key infrastructure [5]. If she had access to the public keys, then she could have applied Rivest et al.'s ring signatures [9] directly to prove that: (a) she stole the secret keys of many parties (this includes insider attacks), (b) she found a way to break the RSA signature scheme (fully, or partly with novel hardware), or (c) the software or hardware implementation is flawed (here we include side-channel attacks and flaws in the underlying operating system). The novel hardware may include quantum computers. We stress that the ring signature may involve public keys from various public key infrastructures from different countries.

Suppose that Jenny instead hacks a server S_j. She can convince the public that one out of many servers $S_1, \ldots, S_k$ has been hacked without revealing which by enforcing certain correlations in the traffic to and from the servers. Even if only deterministic methods are used to derive nonces, signatures, etc, timing of responses can be used to create correlations. She would have to modify the software or configuration of the hacked server, but it could be minimal changes. This causes a loss of trust in *all* servers. The cost of clearing a server park of malware, imaginary or real, is large as is illustrated by the attack on Belgacom [1].

Thus, the cost that Jenny can incur at all levels of abstraction, from theory to implementation, can be large compared to the cost of finding some attack on some system. There are even established price lists for exploits, e.g., the Zerodium Exploit Acquisition Program [13], so stupid Stuart can buy the findings of Jenny at a relatively low price if he has better means to exploit them.

10 Conclusion

Theory of cryptography. The research program of the theory of cryptography is robust. Indeed, cryptography based on generic assumptions such as one-way functions is already resilient against the smear campaign, since (1) there are candidate constructions of one-way functions with very strong trust rooted in general complexity theory, and (2) there are many candidates, so if a particular one-way function is no longer trusted, then it can be replaced trivially. For sufficiently weak assumptions the smear campaign can be tacitly ignored as a rounding error. One cannot hope for more without proving unconditional lower bounds on the complexity of problems.

From an abstract point of view, a combiner is a construction of a novel problem that is hard provided that some of its component problems are hard, but with the *additional benefit of preserving some useful special properties.* This is of course implicit in the concept, and there are combiners that use instantiations of one notion to construct an instantiation of another, but perhaps we should be even more relaxed regarding exactly what special properties the result has. History shows that cryptographers are masters of exploiting the slightest features to their advantage in constructions.

Special-purpose constructions. Many modern constructions are based on computational problems that exhibit special properties, e.g., the discrete logarithm problem and the RSA problem both allow manipulating instances efficiently using group operations, and some elliptic curves even have efficiently computable bilinear maps into finite fields. These are necessary properties for some constructions to be possible at all, or to be reasonably practical.

This fragment of cryptography is fragile in the sense that there are no drop-in replacements for the assumptions used. Thus, attempting to achieve the same results under different assumptions is important, but typically this is still done under assumptions with (other) special properties. The smear campaign shows that this approach may not be as effective to address the lack of trust as one would hope, since it suffices to break any of the special assumptions to cast doubt on all. We think more caution is warranted when these constructions are suggested for real use, in particular in long-lived applications.

Constructions based on problems with special properties are of course important stepping stones to discover constructions based on weaker assumptions, but unfortunately the latter are rarely as efficient and not used in practice. A more constructive view is to consider them to be components of combiners. This would strengthen the role of combiners of all forms in theory and change the view of cryptography based on assumptions with any special features.

Systems. We think that the generalized smear campaign for systems should be taken seriously and be part of real-world risk analyses. Arguably, the most realistic smear campaign is one where a system is hacked and doubt is cast on other systems, protocols, and assumptions, but the most likely culprits are still rarely cryptographic protocols or computational assumptions.

References

1. Der Spiegel. Belgacom attack Britain's GCHQ hacked belgian telecoms firm. http://www.spiegel.de/international/europe/british-spy-agency-gchq-hacked-belgian-telecoms-firm-a-923406.html, Sept. 2013.
2. European network of excellence in cryptology II (ECRYPT II). http://www.ecrypt.eu.org, Nov. 2012.
3. O. Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, New York, NY, USA, 2000.
4. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, Apr. 1988.
5. D. Goodin. Millions of high-security crypto keys crippled by newly discovered flaw. Ars Technica, 2017. https://arstechnica.com/information-technology/2017/10/crypto-failure-cripples-millions-of-high-security-keys-750k-estonian-ids/.
6. D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen. On robust combiners for oblivious transfer and other primitives. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 96–113, 2005.
7. E. Heilman, N. Narula, T. Dryja, and M. Virza. IOTA Vulnerability report: Cryptanalysis of the curl hash function enabling practical signature forgery attacks on the IOTA cryptocurrency, 2017. https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md.
8. N. Koblitz and A. Menezes. Another look at "provable security". *J. Cryptology*, 20(1):3–37, 2007.
9. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 552–565, 2001.
10. D. Shumow and N. Ferguson. On the possibility of a back door in the NIST SP800-90 dual EC Prng. Rump session Crypto '07, 2007. https://rump2007.cr.yp.to/15-shumow.pdf.
11. J. Swihart, B. Winston, and S. Bowe. Zcash counterfeiting vulnerability successfully remediated, 2019. https://z.cash/blog/zcash-counterfeiting-vulnerability-successfully-remediated.
12. Verificatum Mix-Net. http://www.verificatum.org, Jan. 2013.
13. Zerodium. Zerodium Exploit Acquisition Program. https://zerodium.com/program.html, Apr. 2018.