On Cryptographic Assumptions and Challenges

Moni Naor[*]

Weizmann Institute of Science
Rehovot 76100, Israel
naor@wisdom.weizmann.ac.il

Abstract. We deal with computational assumptions needed in order to design secure cryptographic schemes. We suggest a classi£cation of such assumptions based on the complexity of falsifying them (in case they happen not to be true) by creating a challenge (competition) to their validity. As an outcome of this classi£cation we propose several open problems regarding cryptographic tasks that currently do not have a good challenge of that sort. The most outstanding one is the design of an ef£cient block ciphers.

1 The Main Dilemma

Alice and Bob are veteran cryptographers (see Dif£e [15] for their history; apparently RSA [38] is their £rst cooperation). One day, while Bob is sitting in his of£ce his colleague Alice enters and says: "I have designed a new signature scheme. It has an 120 bits long public key and the signatures are 160 bits long". That's fascinating, says Bob, but what computational assumption is it based on? Well, says Alice, it is based on a new trapdoor permutation f_k and a new hash function h and the assumption that after given f_k (but not the trapdoor information) and many pairs of the form $(m_i, f_k^{-1}(h(m_i)))$ it is still hard to come up with a new pair $(m, f_k^{-1}(h(m)))$. So what should be Bob's response? There are several issues regarding these assumptions. For instance, given that the trapdoor permutation and hash function are "new" can they be trusted? But a more bothersome point is whether the assumption Alice uses is really weaker than the assumption "this signature scheme is secure".

So how can we differentiate between the strengths of assumptions and avoid circularity in our arguments? This is the main question we would like to address in this paper and talk and, in particular, to allow answering such doubts and thoughts in a somewhat quantitative manner. We suggest several categories of computational assumptions. Given an assumption it should then be possible to put it into a weaker or stronger class and evaluate it according to this classi£cation. Furthermore, this classi£cation raises many interesting open problems regarding the possibility of basing schemes on assumptions from a weaker class than is currently known.

Designing secure systems is a complex task, even more so than most other design tasks, as there are some malicious entities trying to interfere with the operation of the system. Furthermore, demonstrating that the design is a 'good' one is also not so simple.

---

[*] Incumbent of the Judith Kleeman Professorial Chair. Research supported in part by a grant from the Israel Science Foundation.

First, of course, one should de£ne what 'good' means. The last twenty years have seen great progress in de£ning what a good systems is in the area of Foundations of Cryptography (see Goldreich [24] for a more formal treatment and Bellare and Goldwasser [8] for a more concrete one). For many cryptographic tasks (e.g. signature schemes) we have precise de£nitions of what a good system means i.e. what the attack is and what it means to break the system. In particular:

– The power of the adversary in terms of access to the system is speci£ed (in the context of signature schemes, whether it is an adaptive message attack or not).
– What constitutes failure of the system (in the context of signature schemes, existential forgery means that the adversary comes up with a single message and a valid signature on it that was not explicitly signed by the signer).

For almost any interesting cryptographic task we need to make computational hardness assumptions, i.e. that certain computational tasks require a lot of resources e.g. time and memory*. Furthermore, given the state-of-the-art in Complexity Theory this hardness must be assumed rather than proved.

We deal with hardness assumptions where the adversary is assumed to be limited in its computation time and may succeed with a low probability. The assumptions are stated formally using three parameters n, t and ϵ as follows . Let n be the instance size (which can be thought of as the security parameter), t the upper bound on the time available to the adversary and ϵ the probability of success. We assume that there is some underlying distribution on instances (this could be for instance the choice of a key in signature scheme). For an assumption to be 'hard' we require that for instances of size n no adversary whose run time is bounded by t can succeed (where the notion of succeeds depends on the problem type) with probability better than ϵ. The probability of success is taken over both the distribution of the instance and the coin ¤ips of the adversary. The upper bound on the time t and upper bound on success probability ϵ are a function n. Note that this is a concrete parameters treatment, rather than an asymptotic one, but it is more a matter of style rather than substance. We could have speci£ed our classes in an asymptotic manner.

In general when we are proving that a System X is secure based on an Assumption A, then what we show is that either (i) X is secure or (ii) Assumption A is false. However, the problem with this sort of conditional statement is that it might be hard to demonstrate that A is false (especially since it's collapse is relatively rear), so we don't have good guidelines to decide whether to use the scheme or not.

A proof of security is interesting if (but not necessarily only if) a system whose security requirement is (apparently) not ef£ciently falsi£able is based on an assumption that is ef£ciently falsi£able.

Paper plan: In Section 1.1 we list four assumptions representing the various types of assumptions we deal with in this paper. In Section 2 we present our main thesis regarding assumption and challenges. Section 3 we discuss common cryptographic tasks and

* Notable exceptions to tasks that must rely on the adversary's computational limitation are encryption with a one-time pad, authentication with a one-time key and multi-party computation where a majority of the players are honest and private lines are available.

where they naturally reside. In Section 4 we provide a collection of open problems that our methodology raises. Finally, in Section 5 we provide some caveats regarding our classi£cation.

1.1 Examples of Assumptions

We now describe four assumptions that will be used to demonstrate the new concept introduced in the paper. The assumptions we use involve factoring, RSA and Discrete log type. The £rst two are well known while the last two are more esoteric. Let

$$Z^{(2)}(n) = \{N = pq | p, q \text{ are } n\text{-bit primes}\}.$$

Factoring: Consider the assumption that factoring is hard: Let the input be an N chosen uniformly at random from $Z^{(2)}(n)$ (denoted $N \in_R Z^{(2)}(n)$) and the computational task the adversary $\mathcal{A}$ is faced with is to £nd n-bit primes p and q such that $N = p \cdot q$. The (n, t, ϵ) hardness of factoring assumption is: no t-time algorithm $\mathcal{A}$ satis£es:

$$\Pr[\mathcal{A}(N) = (p, q) \text{ where } p \cdot q = N)] > \epsilon.$$

The probability is over the choice of N and the coin ¤ips of $\mathcal{A}$. Note that it is easy to verify a solution, and since we have good primality algorithms (either probabilistic or deterministic) we can even check that N is indeed in $Z^{(2)}(n)$.

The RSA Assumption: The input distribution is a random $N \in_R Z^{(2)}(n)$, e that is relatively prime to $\phi(N)$ and $s \in_R Z_N^*$. The computational task the adversary $\mathcal{A}$ is faced with is to £nd a such that $a^e = s \bmod N$. The (n, t, ϵ) hardness of RSA assumption is: no t-time algorithm $\mathcal{A}$ satis£es:

$$\Pr[\mathcal{A}(N, e, s) = a \text{ where } a^e = s \bmod N] > \epsilon.$$

Difference RSA assumption: The difference RSA assumption deals with the hardness of £nding two RSA preimages such that the difference of their images is a given quantity D, even when many examples with this property are given. The input distribution is a random $N \in_R Z^{(2)}(n)$, e (an RSA exponent) and $D \in_R Z_N^*$. The adversary $\mathcal{A}$ has access to sequence of $m-1$ pairs (x_i, y_i) s.t. $x_i^e - y_i^e = D \bmod N$ where $\mathcal{A}$ chooses x_i. The adversary should £nd new (x_m, y_m) (not in $((x_1, y_1), (x_2, y_2) \ldots (x_{m-1}, y_{m-1}))$ such that $X_m^e - Y_m^e = D \bmod N$.

The (n, t, ϵ) Difference RSA assumption: *no* t-time algorithm $\mathcal{A}$ that is given (N, e, D) as above and may access is given input pairs with the £rst value being his choice satis£es:

$$\Pr[\mathcal{A}(N, e) = (x_m, y_m) \text{ s.t. } x_m^e - x_m^e = D \bmod N \text{ and } (x_m, y_m) \text{ is new}] > \epsilon.$$

Knowledge of Exponent [26] Let Q and P be primes such that $Q|P-1$, and g a generator of a subgroup of size $Q \bmod P$. The input distribution is a random $h = g^x \bmod P$ for $x \in_R [Q]$ (i.e. h is a random element in the group generated by g; the exponent x is secret). The adversary tries to come with h_1 and h_2 where $h_1 = g^z \bmod P$ and

$h_2 = h^z \bmod P$ for some $z \in [Q]$. The point of the assumption is that the only concievable way of doing so is to £rst pick z and then exponentiate g and h with z; therefore any algorithm that outputs such a pair really 'knows' z. The assumption is for any t-time algorithm $\mathcal{A}$ there is a t-time algorithm $\mathcal{A}$ s.t:

$$|\sum_z \Pr[\mathcal{A}(g,h) = (g^z, h^z)] - \sum_z \Pr[\mathcal{A}(g,h) = (z, g^z, h^z)]| < \epsilon$$

Note that this is essentially a proof of knowledge, but unlike the usual de£nition of proof of knowledge there is no black-box knowledge extractor here (since there is no interaction), but rather the assumption is that it is always possible to output the same distribution as the original one but with z as well.

An observant reader can detect dependencies between these assumptions. Speci£cally, RSA assumes Factoring and Difference-RSA assumes RSA. But are there more fundamental differences? these assumptions will be used to exemplify the various categories we introduce.

2 Thesis

In order to be able to evaluate whether an assumption is true or not it must be falsi£able, i.e. there must be a (constructive) way to demonstrate that it is false, *if this is the case*. Furthermore, the complexity of checking the refutation of the assumption is of interest and should be a major consideration in how acceptable the assumption is.

We are mostly interested in falsi£cation by challenge, where a public challenge is advertised. If the assumption is false, then it is possible to solve the challenge (and the solution can be veri£ed) in time related to the time it takes to break the assumption. Falsi£cation by challenge embodies a fundamental principle that underlies the scienti£c investigation of cryptography: that public evaluation and discussion of cryptographic schemes enhances their security and is essential for the acceptance of any scheme**.

We de£ne the complexity of falsifying an assumption from the perspective of the protocol designer or the one who should evaluate how good is the design. Therefore we will be interested in

- How dif£cult it is to generate a random challenge and in particular what is the size of the challenge (which should be small).
- How dif£cult it is to verify that a proposed solution to the challenge is indeed a correct one. Ideally it should involve computing a simple function or even simply comparing the result to a precomputed solution.
- Is the veri£cation of the solution public, i.e. simply a function of the public challenge and the solution or is secret information needed, i.e. the challenge is generated together with some additional input that can help check the solution. The former is preferable of course, since the veri£cation is public and there is no need for a trusted party (that would leak the solution...).

** Although this principle can be traced back to Kerckhoffs in 1883 [29], it has been executed in earnest only since the 1970's.

An important entity in our world view is the *falsi£er*. The role of the falsi£er it to solve the challenges published by the protocol designer (or someone wishing to use it). In other words, what the adversary is to the assumption the falsi£er is to the challenge. There are two main differences between them, one is that an assumption might be interactive (though none of those in Section 1.1 are interactive) whereas a challenge is by de£nition not, but is of the "send-and-forget" type. Another difference is that if an assumption is false, then the probability of success might be rather small, just over ϵ, whereas a falsi£er needs a much better probability of success. The protocol designer would like to encourage would be falsi£ers (or falsi£er designers, keep in mind that the acceptance of a protocol is a social process), so the probability of success should be high, close to 1, (if the assumption A is false).

2.1 The categories of falsi£cation

In this section we present our categories of falsi£cation via a challenge. As mentioned above, a major consideration is the size of the challenge d and the run time of the veri£er V that should verify the solution y to the challenge d. A challenge is speci£ed by D_n a distribution of challenges of size n. In general given a challenge d chosen according to D_n there should be at least one solution y such that V accepts y (as a solution to d). One additional parameter we need is δ, which is an upper bound on the failure of the challenge, i.e. even if the original assumption A is false, there might be a chance (bounded by δ) that the challenge will not be solvable.

So what we want is a one-to-one correspondence between the adversary for breaking A and the falsi£er for the challenge. If there is a an adversary that runs in time t and succeeds with probability better than ϵ in breaking A, then there should be a falsi£er that runs in time t' which is some (£xed) polynomial in t and $1/\epsilon$ and succeeds with high probability. This assures us that instead of looking for ways to break A (where the reward is ϵ) we may as well look for ways for solving the challenge. Similarly, we would like the challenge to be non-trivial, i.e. if it can be solved with probability γ, then the original assumption A should also be solvable, with probability which is at least polynomial in γ and $1/n$.

We present three categories of assumptions which we call *ef£ciently falsi£able, falsi£able* and *somewhat falsi£able*. The differences between them are based mainly on the time to verify the solution. Since the process we have in mind regarding the challenges does not assume that the one who suggests the falsi£er and the protocol designer belong to the same body or entity, it is important that the interaction between them be as simple as possible. Therefore ideas such as simulating the falsi£er (running his program) should be used only as a last resort.

An adversary $\mathcal{A}$ to Assumption A with parameters (n, t, ϵ) means that $\mathcal{A}$ working in time at most t can break the assumption on instances of size n with probability better than ϵ.

De£nition 1 (Ef£ciently falsi£able (EF)). *An* (n, t, ϵ) *assumption* A *is* ef£ciently falsi£able *if there is a distribution on challenges* D_n *and a veri£cation procedure* $V : \{0,1\}^* \times \{0,1\}^* \mapsto \{$*accept, reject*$\}$ *such that*

– *Sampling from D_n can be done ef£ciently, in time polynomial in* $n, \log 1/\epsilon$ *and* $\log 1/\delta$.
– *The run time of V is polynomial in n,* $\log 1/\epsilon$ *and* $\log 1/\delta$.
– *If the assumption A is false then there exists a falsi£er $\mathcal{B}$ that for a $d \in_R D_n$ £nds a y for which $V(d, y)$ is 'accept' with probability at least $1 - \delta$ and the run time of $\mathcal{B}$ is t' which is polynomial in the run-time of $\mathcal{A}$ as well as* $\log 1/\epsilon$ *and* $\log 1/\delta$ *and* n.
– *If there is a falsi£er $\mathcal{B}$ that solves random challenges $d \in_R D_n$ in time t with probability at least γ, then there is a (t, ϵ) adversary $\mathcal{A}$ for breaking the original assumption A where t' and ϵ' are polynomially related to t and γ.*

Which assumptions are ef£ciently falsi£able? An interesting class of assumptions is those that are based on a random self reducible problem, i.e. given any instance it is possible to generate a random instance (or instances) whose solution would yield one for the original. One such problem is discrete log (and related problems such as Dif£e-Hellman). Similarly the Ajtai and Dwork $[1, 2]$ reductions also yield ef£ciently falsi£able assumptions. The example of factoring is a little less straightforward and is quite instructive:

Factoring is ef£ciently falsi£able (EF): we £rst attempt to show that factoring is EF by choosing the challenge distribution D_n as random $N \in_R Z^{(2)}(n)$. The desired solution is p and q such that $N = p \cdot q$. However, there is a problem: suppose that there are a fraction of 2ϵ of $Z^{(2)}(n)$ that are very easy to factor (denote them by Z'. Then the assumption that factoring is (t, ϵ)-hard is false. However the probability that the challenge will be chosen from Z' is small, 2ϵ, whereas the probability that a falsi£er should succeed in solving the challenge should be large, $1-\delta$. So the problem is that the bad set is too sparse and we need 'easiness' ampli£cation (which is related to the much used hardness ampli£cation). The idea is to allow sampling roughly $1/\epsilon$ samples from $Z^{(2)}(n)$. First note that $Z^{(2)}(n)$ is not so sparse with respect to all $2n$-bit numbers. So if we sample polynomially many $2n$-bit numbers we are bound to hit $Z^{(2)}(n)$ many times. We employ standard technique for ampli£cation using pair-wise independent hashing. Let H be a family of pairwise independent hash functions where for $h \in H$ we have $h : \{0, 1\}^n \mapsto \{0, 1\}^{n-2\log n+2\log\epsilon}$. One additional property we need from H is that it will be easy to invert h on a given point c (standard construction enjoy this property). The challenge d is speci£ed by (h, c) where $h \in_R H$ and $c \in_R \{0, 1\}^{n-\log n+2\log\epsilon}$. A solution is p and q such that $h(p \cdot q) = c$ and p and q are n-bit primes. This gives us a £xed δ, so to get any δ we repeat it $\log 1/\delta$ times and the challenge is to solve one instance.

The claim is that if more there is an adversary $\mathcal{A}$ that succeeds in factoring with probability better than ϵ, then there is a falsi£er that succeeds with probability $1 - \delta$. The falsi£er simply runs $\mathcal{A}$ on all $N \in h^{-1}(c)$ and checks the results for primality.

When relaxing the notion of 'ef£ciently falsi£able' to 'falsi£able' we change the requirement that veri£cation time will be proportional to (polynomial in) to that of polynomial in $1/\epsilon$. In particular the challenge size can be much larger now.

De£nition 2 (Falsi£able). *An (n, t, ϵ) assumption A is* falsi£able *if everything is as in De£nition 1, except that the run time of sampling D_n and V may depend on $1/\epsilon$ (rather than $\log 1/\epsilon$).*

RSA is falsi£able: The RSA problem is similar in nature to factoring. Therefore it is simple to make it into a falsi£able assumption: sample $1/\epsilon$ times from $Z^{(2)}(n)$ and choose a random prime to generate a pair (N_i, e_i). Choose a random s and publish the results as a challenge. A solution is any y such that there exits an i such that $y^{e_i} = s \bmod N_i$. However, there does not seem to be a simple way of making the RSA Assumption ef£ciently falsi£able, since we don't know how to obliviously sample from the domain of the function. The reason it did not hurt with the factoring example is that the solution £ltered out those that were not of the proper form.

De£nition 3. *An (n, t, ϵ) assumption A is* somewhat falsi£able *if everything is as in De£nition 1 accept that the run time of V may depend on $1/\epsilon$ (rather than $\log 1/\epsilon$) and the run time of $\mathcal{B}$. In particular this means that V may simulate $\mathcal{B}$.*

Difference RSA is somewhat falsi£able: When considering the problem of Difference RSA, the problem in making it falsi£able is that the amount of information the attacker is allowed to obtain is large and is not bounded by a £xed polynomial in n and $1/\epsilon$. Furthermore the adversary can choose it dynamically. So there does not seem to be a publishable challenge. Instead what the veri£er can do is to simply run the program of the supposed falsi£er (each time with a self chosen $N \in_R Z^{(2)}(n)$, so that it is possible to answer the query). This should be executed $1/\epsilon$ times and if in at least one of them the program succeeds in £nding a different pair than was given to it we conclude that the Difference RSA assumption is false.

None of the above: £nally, there seem to be assumption that do not fall into any of the categories mentioned above. These includes assumptions where it is not so clear whether the adversary has won or not. This include the *Knowledge of an exponent* problem, since given a program that outputs consistent pairs (where both h_1 and h_2 have the same exponent) it is not clear that it is hard to extract the exponent from it. What seems to be the problem in making this assumption even somewhat falsi£able is the order of quanti£ers in the speci£cation (for all program there exists an extract).

3 Some cryptographic tasks and their classi£cation

We now list a few cryptographic tasks and discuss in what class the assumption "this scheme is a secure implementation of the task" lies (at least to the best of our knowledge, since we do not how to demonstrate that an assumption is not in a weaker class than what we know how to show).

One-way permutations: given a function $f : \{0,1\}^n \mapsto \{0,1\}^n$ is it a way permutation, i.e. no t time algorithm can £nd x for a random $y = f(x)$ with probability better than ϵ. We claim that the assumption "f is a one-way permutation" is ef£ciently falsi£able. This is true for similar reasons as factoring: choose a set of challenges speci£ed as a range of a function. Let H be a family of pairwise independent hash functions, for $h \in H$ we have $h : \{0,1\}^n \mapsto 0,1^{n-\log n+2\log\epsilon+log\delta}$. The challenge is speci£ed by $h \in_R H$ and $c \in \{0,1\}^{n-\log n+2\log\epsilon+log\delta}$. A solution is x such that $f(x) = y$ and $h(y) = c$.

Note that we are not checking the permutation part, i.e. that f is indeed $1-1$ and length preserving (this is an unconditional property and could be proved irrespective of the developments in Complexity Theory.

Regarding an assumption of the form "f is a one-way function" (as opposed to a permutation), then we do not know whether it is necessarily an ef£ciently falsi£able assumption, since it is not clear how to sample from the domain of the function f (without explicitly computing the values). Therefore the best we can say is that it is a falsi£able assumption by generating $1/\epsilon$ values $y_i = f(x_i)$ and the challenge is to solve at least one of them.

Pseudo-random Sequences: We say that a function $G : \{0,1\}^m \mapsto \{0,1\}^n$ is a cryptographically strong pseudo-random sequence generator if it passes all polynomial time statistical tests. More concretely that it is (t,ϵ)-pseudo-random if no test $\mathcal{A}$ running in time t can distinguish outputs of G from random string of length n with advantage greater than ϵ. We do not know how to make the assumption "G is a pseudo-random sequence generator" into an ef£ciently falsi£able one. However, we can show that it is a falsi£able one. The idea is to publish a bunch of pairs of strings (random, output of G) in random order and the challenge is to distinguish correctly for many (more than half) of them between the truly random part and the $G(x)$ part. More speci£cally, the challenge consists of "many" (polynomial in $1/\epsilon$ and $\log 1/\delta$) pairs $\{(x_i, y_i)\} \in \{0,1\}^{2n}$, where one element in the pair is random and one pseudo-random and the solution should identify correctly $1/2 + \epsilon/2$ of the pairs. Note that here we are assuming that V has some secret information - the correct ordering of the pairs (which is in violation of De£nition 1).

Pseudo-random functions, Block Ciphers and MACs We now deal with three types of keyed functions where during a learning phase the adversary can query them at many points of its choice and then has to pass a test related to a point it chooses but one that has not been queried before. In the case of a pseudo-random function [22] the adversary has to distinguish between a value of a given point and a truly random string, in the case of a MAC or an unpredictable function (the terminology is from [34]) it has to guess its value on the point. For block copers, or equivalently pseudo-random permutation [30, 33] the function is a permutation on the inputs and the adversary can ask encryption or decryption queries (to go forward or backwards in the permutation). It then has to distinguish the given value from random. Given the interactive nature of the queries of the adversary and given that the adversary can choose both the queries and the test point, there does not seem to be a way to make a challenge that is only of size proportional to $\log 1/\epsilon$ and n, so the best we know how to do is simulate the adversary (and then we can effectively test whether it has succeeded). This makes all assumptions of the form "F_k is a family of (pseudo-random functions—pseudo-random permutations—unpredictable functions)" into somewhat falsi£able ones.

On the other hand there are constructions of pseudo-random functions that are ef£ciently falsi£able, for instance the Goldreich, Goldwasser and Micali construction [22] when the underlying pseudo-random generator is based on a one-way permutation, or the constructions of Naor, Reingold and Rosen based on factoring and Decisional Dif£e-Hellman [32, 35]. Similarly for pseudo-random permutations (block-ciphers).

Signatures Schemes: The acceptable notion of a good signature scheme is that of an existentially unforgeable under an adaptive message attack [23]. A signature scheme satis£es this notion if no t-time adversary has probability of success better than ϵ of breaking the scheme where:

- The access to the system consists of adaptively choosing messages m_i and receiving valid signature s_i for $q-1$ rounds ($q \leq t$).
- The adversary tries to £nd a pair (m_q, s_q) where $m_q \notin \{m_1, m_2, \ldots m_{q-1}\}$ and s_q is a valid signature on s_q.

As far as we know the requirement that a signatures scheme be existentially unforgeable under an adaptive chosen message attack is not ef£ciently falsi£able of even just plain falsi£able, for the same reason as pseudo-random functions are not: the attacker can adaptively choose both the queries and the test. Again it is a somewhat falsi£able assumption, since by simulating the adversary it is possible to check whether it has succeeded or not.

Encryption Schemes : the de£nitions of a good encryption scheme come in many shapes and sizes, depending on whether the scheme is shared or public key, the type of attack (known plaintext, chosen plaintext, chosen ciphertext of various shades) and what it means to break the system (semantic security, non-malleability). (See [16] and [5] for more details.) All these requirement are somewhat falsi£able, since it is possible to simulate the adversary and check whether it succeeds or not and hence evaluate the probability of success. On the other hand they do not seem to be in any weaker class. There is one exception which is falsi£able: when the scheme is public key, the messages are encrypted bit-by-bit, the attack is chosen plaintext (the weakest one relevant in this context) and the security requirement is semantic security.

Zero-knowledge : the requirement that a proof system be zero knowledge is that for any behavior by the veri£er denoted by $\mathcal{V}^*$ there should be a simulator $\mathcal{S}$ that outputs a simulated conversation between the prover and the veri£er. The distributions on the conversations between $(\mathcal{P}, \mathcal{V}^*)$ and the outputs of $\mathcal{S}$ should be indistinguishable. This requirement does not seem to be even somewhat falsi£able due to the order of the quanti£ers: suppose that a falsi£er comes up with a seemingly bad $\mathcal{V}^*$, how do we (the protocol designers) know that there is not simulator for $\mathcal{V}^*$. This is especially acute in case of non block-box simulation [4].

In contrast, the property of Witness Indistinguishable is somewhat falsi£able.

4 Challenging Problems

One of the main points of this paper is that the classi£cation just proposed yields many open problems. For any cryptographic task the question is whether it is possible to base it on an assumption from a weaker class than where the tasks lies “naturally” (i.e. where the assumption “this system implements the task in a secure manner” lies). One can view many of the signi£cant and interesting results in cryptography this way, basing a system on an assumption from a weaker class; e.g. any cryptographic scheme based on factoring (and there are many examples of such) is based on an ef£ciently falsi£able assumption, no matter where it naturally lies.

We now list a few such problems. In most of the cases there is an implementation based on random oracles [27, 6], i.e. the analysis is based on the assumption that a certain function behaves as a random function. Once the random function is substituted with a concrete one we may phrase an assumption on what is needed from the function. However, usually it is often not simpler than “this system is secure.”

Ef£cient Block ciphers: common block ciphers such as DES or AES (Reijndael) are several orders of magnitude faster than schemes based on the hardness of factoring or any other scheme that can be based on an ef£ciently falsi£able assumption. On the other hand they do not seem to yield any assumption that is simpler than “this scheme is secure”. So the most outstanding problem our methodology raises is whether it is possible to come up with an ef£cient proposal to a block cipher, competitive with the state-of-the-art, together with a meaningful succinct challenge , i.e. a block cipher based on an ef£ciently falsi£able assumption.

Signatures: while there have been many proposals of signatures schemes based on ef£ciently falsi£able or just falsi£able assumption, they are mostly less ef£cient than ones based on random oracles [7], but some of them give the latter a run for their money [14, 21]. However, there have been several parameters where no non random oracles constructions are known:

Short signatures: The schemes with the shortest signature construction known are based on Weil Pairing and random oracles, and get to less than 200 bits per signature [10]. These sort of succinct signatures are important when the signature is carried over a very low bandwidth channel, e.g. a human saying it over the phone. So the question is whether there exists an ef£ciently refutable assumption allowing such short signatures.

Low overhead signatures There are methods that allow signatures with a short additional string, mostly by using the signatures itself to store the string (signatures with recovery). This is important to lower the overall bandwidth overhead of sending a signature. These schemes are based on RSA or Rabin plus a random oracle [7]. The question is whether it is possible to get to this overhead with overhead signatures which are based on a falsi£able assumption.

Discrete log signatures To the best of our knowledge no discrete log type signature scheme (El Gamal, Schnorr) has been shown secure without the help of random oracles.

Low exponent veri£cation RSA or Rabin provide signatures with extremely ef£cient veri£cation. Is it possible to obtain such schemes based on a falsi£able assumption?

Identity Based Encryption: In 1984 Adi Shamir [40] suggested the notion of an identity based encryption (IBE), where a short public key determined implicitly the public keys of all participants (in conjunction with their 'identity') and a center that know a secret can provide them with the corresponding secret key. One of the more interesting developments in the last two or three years has been the £rst proposals for IBEs by Boneh and Franklin [9] and Cocks [13]. While the scheme of [9] is provided with a rigorous analysis, as far as we can tell one does not yield an ef£cient falsi£able assumption from it, so the open question is how to obtain a scheme based on such an assumption.

Blind Signatures and anonymity A blind signature is one where the signer does not know what it is signing yet it is impossible to generate more signed messages than the signer provided. This notion, proposed by Chaum [11] has many application from untraceable electronic money [12] to fairness in two party computation [36]. It has been analyzed in the random oracle model [37] and there is a highly interactive (between the signer and signee). There is also construction with no random oracles based on secure function evaluation [28]. So the question is whether a two round signature scheme (request and response) is possible based on ef£ciently falsi£able assumption. This is also true for other anonymity preserving tools such as deniable group signature [39] (based on RSA and random oracles vs. the interactive version given in [31] (based on any good encryption scheme).

Three round zero knowledge: One of the outstanding open problems regarding zero-knowledge proofs is whether it is possible to obtain a three round (move) protocol with low probability of error. Hada and Tanaka [26] introduced the assumption of "Knowledge of Exponent" in order to resolve this issue, but as far as we know this assumption is not even somewhat falsi£able. So the question is whether it is possible to obtain such a protocol with an ef£ciently falsi£able assumption. See [19, 20] for recent work on the subject.

Incompressible functions: Let $f : \{0,1\}^n \mapsto \{0,1\}^m$ be an expanding function. Suppose that one party Alice knows x, and wants the other party Bob to learn $f(x)$ *without revealing more information than necessary about* x. Alice could of course simply send $f(x)$ to Bob, but the goal is to have a low communication protocol. A function f is called *incompressible* if any $o(m)$ bit message enabling the computation of $f(x)$ reveals x. This property was introduced by [17] in the context of preventing the illicit distribution of keys for broadcast encryption. They also conjectured that the function $f(x) = g^x \bmod P \circ g^{x^2} \bmod P \circ \ldots \circ g^{x^\ell} \bmod P$ is incompressible. However, as far as we can tell the assumption that a function f is incompressible is not even somewhat falsi£able. So the open problem is to come up with a candidate for an incompressible function where the assumption that it is incompressible is ef£ciently falsi£able.

Moderately hard functions Moderately hard functions, where the time it takes to evaluate them is neither too small nor too large have many applications in cryptography and computer security, from abuse prevention (See [18]) to zero-knowledge [19]. The question is whether it is possible to incorporate assumptions regarding moderately hard functions into this framework.

Physical security: We have dealt so far with the usual mathematical abstract of a distributed system and ignored to a large degree the physical realization, however, as we have seen in the last few years often times "physical" attacks and issues such as: power consumption, tamper resistance, fault Resistance and timing make the implementation very vulnerable (see [3]). So ideally we would like to base security on simple assumption that can be falsi£ed similarly to computational assumption.

This is true also for humans. It is often said that the human user is the weakest link in the security of a system. So again we would like to incorporate human ability considerations into the security design in a rigorous manner and in particular make falsi£able assumptions about human users ability and prove the security of a system based on these assumptions?

5 Caveats

While we believe that the classi£cation we have presented is widely applicable and can clarify many issues, it is not a panacea and many important issues are not handled by it.

Implication not preserved by classi£cation: If Assumption A implies Assumption B and A is ef£ciently falsi£able we cannot conclude that B is ef£ciently falsi£able as well. The reason is that we cannot simply use the challenges for A to check B, since a falsi£er for A is not necessarily one for B (recall that we should be able to use it to break A).

Classi£cation does not highlight all major results in cryptography: One of the most outstanding results in Foundation of Cryptography is the HILL (Håstad, Impagliazzo, Levin and Luby) [25] stating that it is possible to construct a pseudo-random sequence generator from any one-way function. On the other hand, according to our classi£cation both the assumption and the outcome are in the class of falsi£able tasks (though the latter is there only when the veri£er has secret information). So such a major result is not highlighted by our classi£cation. This true for many general results, since there is no particular payoff for a general result. In particular the issue may be that falsi£ability provides only a partial order between assumptions (given the we have only four categories then clearly this is not a linear scale).

History of investigating an assumption is not accounted for. A major reason to adopt an assumption is the history of computational attempts to resolve it. This is completely ignored in our framework, but on the other hand our framework yields the opportunity to create such a history for an assumption or an area.

Do not verify run time if the one taking up the challenge has more computational resources than the protocol designer assumes then he may simply break the challenge "in brute force" (without violating the formal statement). But this can be seen as violating another implicit assumption "that t is an infeasible amount of total computational time available to the attacker."

Acknowledgements

I have had many interesting conversation with my colleagues regarding the issues outlined in the paper. In particular I would like to thank Dan Boneh, Cynthia Dwork, Benny Pinkas and Steven Rudich. I thank Dalit Naor for many comments regarding the writeup.

References

1. M. Ajtai, *Generating Hard Instances of Lattice Problems*, Proceedings of the 27th ACM Symposium on Theory of Computing, 1996, 99–108.
2. M. Ajtai and C. Dwork, *A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence,* Proceedings of the 28th ACM Symposium on Theory of Computing, 1997, pp. 284–293.
3. R. Anderson, **Security Engineering: A guide to Building Dependeable Distributed Systems**, Wiley, 2001.
4. B. Barak, *How to Go Beyond The Black-Box Simulation Barrier*, Proc. of the 42nd IEEE Symposium on the Foundation of Computer Science, 2001.
5. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. *Relations among notions of security for public-key encryption schemes,* Advances in Cryptology – CRYPTO'98, Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 26–45.
6. M. Bellare and P. Rogaway, *Random oracles are practical: A paradigm for designing ef£cient protocols Authors*, ACM Conference on Computer and Communications Security, 1993, pp. 62–73.
7. M. Bellare and P. Rogaway, *The Exact Security of Digital Signatures - HOw to Sign with RSA and Rabin*, Advances in Cryptology – Eurocrypt 1996: pp. 399–416
8. S. Goldwasser and M. Bellare, Lecture Notes on Cryptography, 1996, available `http://philby.ucsd.edu/cryptolib/BOOKS/gb.html`
9. D. Boneh and M. Franklin, *Identity Based Encryption from the Weil Pairing*, Advances in Cryptology - CRYPTO 2001, Lecture Notes in Computer Science 2139, Springer, 2001, pp. 213–229.
10. D. Boneh, B. Lynn and H. Shacham, *Short signatures from the Weil pairing*, Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science, Springer, pp. 514–532.
11. D. Chaum. *Blind signatures for untraceable payments*, Advances in Cryptology - Proceedings of Crypto 82, Plenum Press, New York and London, pp. 199–203. 1983
12. D. Chaum, A. Fiat and M. Naor, *Untraceable Electronic Cash* Advances in Cryptology – CRYPTO 1988, ecture Notes in Computer Science, Springer, pp. 319–327.
13. C. Cocks, *An identity based encryption scheme based on quadratic residues*, Cryptography and Coding, Lecture Notes in Computer Science 2260, Springer, 2001, pp. 360–363.
14. R. Cramer and V. Shoup, *Signature Schemes Based on the Strong RSA Assumption*, ACM Conference on Computer and Communications Security, 1999, pp. 46–51.
15. W. Dif£e, *The First Ten Years of Public Key Cryptography*, in **Contemporary Cryptography The Science of Information Integrity**, edited by G. J. Simmons, IEEE Press, 1992.
16. D. Dolev, C. Dwork and M. Naor, *Non-malleable Cryptography*, Siam J. on Computing, vol 30, 2000, pp. 391–437.
17. C. Dwork, J. Lotspiech and M. Naor, *Digital Signets: Self-Enforcing Protection of Digital Information*, Proceedings of the 27th ACM Symposium on Theory of Computing, 1996, pp. 489–498.
18. C. Dwork and M. Naor, *Pricing via Processing, Or, Combatting Junk Mail*, Advances in Cryptology – CRYPTO'92, Lecture Notes in Computer Science No. 740, Springer, 1993, pp. 139–147.

19. C. Dwork and M. Naor, *Zaps and their Applications*, Proceedings of the IEEE 41st Annual Symposium on the Foundations of Computer Science, 2000, pp. 283–293.
20. C. Dwork and L. Stockmeyer,
21. R. Gennaro, S. Halevi and T. Rabin, *Secure Hash-and-Sign Signatures Without the Random Oracle*, Advances in Cryptology – Eurocrypt'99 Proceeding, Lecture Notes in Computer Science, Springer, 1999, pp. 123-139
22. O. Goldreich, S. Goldwasser and S. Micali, *How to Construct Random Functions,* JACM 33(4), 1986, pp. 792–807.
23. S. Goldwasser, S. Micali and R. Rivest, *A secure digital signature scheme*, SIAM J. on Computing 17, 1988, pp. 281–308.
24. O. Goldreich, *On the Foundations of Modern Cryptography,* Advances in Cryptology – Crypto'97 Proceeding, Lecture Notes in Computer Science, Springer, 1997. See `http://www.wisdom.weizmann.ac.il/˜oded/foc.html`
25. J. Håstad, R. Impagliazzo, L. A. Levin and M. Luby, *A Pseudorandom Generator from any One-way Function*, SIAM J. Computing, 28(4), 1999, pp. 1364–1396.
26. S. Hada and T. Tanaka, *On the Existence of 3-Round Zero-Knowledge Protocols,* Advances in Cryptology – Crypto'98 Proceeding, Lecture Notes in Computer Science No. 1462, Springer, 1998, pp. 408–423.
27. R. Impagliazzo and S. Rudich, *Limits on the Provable Consequences of One-Way Permutations*, STOC 1988.
28. A. Juels, M. Luby and R. Ostrovsky *Security of Blind Digital Signatures*, Advances in Cryptology – Crypto'97 Proceeding, Lecture Notes in Computer Science, pp. 150-164
29. A. Kerckhoffs, *La Cryptographie Militaire*, Journal des Sciences Militaires, 1883, pp. 5-38. Available at `http://www.cl.cam.ac.uk/usres/fapp2/kerckhoffs/`.
30. M. Luby and C. Rackoff, *How to construct pseudorandom permutations and pseudorandom functions*, SIAM J. Computing, vol. 17, 1988, pp. 373-386.
31. M. Naor, *Deniable Ring Authentication*, Advances in Cryptology – CRYPTO 2002 Proceeding, Lecture Notes in Computer Science 2442, Springer, 2002, pp. 481–498.
32. M. Naor and O. Reingold, *Number-theoretic constructions of ef£cient pseudo-random functions*, Proceedings of the IEEE 38th Symposium on the Foundations of Computer Science, Oct. 1997, pp. 458–467.
33. M. Naor and O. Reingold, *On the Construction of Pseudo-Random Permutations: Luby-Rackoff Revisited*, Journal of Cryptology, vol 12, 1999, pp. 29–66.
34. M. Naor and O. Reingold, *From Unpredictability to Indistinguishability: A Simple Construction of Pseudo-Random Functions from MACs*, Advances in Cryptology – Crypto'98 Proceeding, Lecture Notes in Computer Science No. 1462, Springer, 1998, pp. 267–282.
35. M. Naor, O. Reingold and A. Rosen. *Pseudo-Random Functions and Factoring.* Siam J. on Computing Vol. 31 (5), pp. 1383–1404, 2002.
36. B. Pinkas, *Fair Secure Two-Party*, Advances in Cryptology - Eurocrypt'2003, Lecture Notes in Computer Science, Springer, 2003, pp. 87–105.
37. D. Pointcheval, J. Stern, *Security Arguments for Digital Signatures and Blind Signatures*, J. of Cryptology 13(3): pp. 361–396, 2000.
38. R. L. Rivest, A. Shamir, and L. M. Adleman, *A method for obtaining digital signature and public key cryptosystems*, Communications of the ACM, vol. 21, Feb 1978, pp. 120–126.
39. R. L. Rivest, A. Shamir, and Y. Tauman, *How to Leak A Secret*, Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248, Springer, pp. 552–565.
40. A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Advances in Cryptology - CRYPTO'84, Lecture Notes in Computer Science, Springer, 1985, pp. 47–53.