Proofs by Reduction

Florian Weber

Since many answers to especially exercise 4b on the third exercise sheet didn't get many points, with some of them suggesting a lack of understanding what you are supposed to do, I've decided to write up a more complete example in which I will try to emphasise the main problems that I found.

We can get to the first one before we even discuss any techniques: Exercise 4b is **completely unrelated** to exercise 4a! While you are supposed to find a (rather trivial) attack in 4a, 4b asks you to convert an arbitrary attack about which you know absolutely nothing, except that it will with probability p find a preimage to one of t hashes, into one that finds a preimage with probability $\frac{p}{t}$ while running in a similar time. Your proof still needs to hold even if p is zero or the attack takes exponential or constant time.

Any more than that would however spoil the exercise for the master students that still have to do it, so I will instead present a different example.

In exercise 3 quite a few of you talked about *existence* of a collision. To be clear: Collisions *exist* for every non-injective function, what we care about in cryptography is how hard it is to *find* them. I didn't subtract any points for that if your proofs were otherwise correct, but please try to avoid doing this in the future.

Finally, please note that the following is *way* more detailed than what is expected from you for the exercises: I'm focusing on *explaining* the techniques used for the proofs here and merely use the provided examples to demonstrate aspects of that. A proof that just tries to prove a statement can (and should) be much shorter, similar to what Andreas Hűlsing showed you in the lecture.

Games and Security Definitions

When working with reductions, we usually define our security notions as so called **games**. A game (usually) consists of two parties: A *challenger* who, as the name suggests, creates a challenge and an *adversary* or *attacker* who tries to break it. If you don't know where to start with a proof it is always a good idea to make sure that you fully understand the games that you will have to work with.

A very simple example for a game would be this:

- The challenger samples a random bit b and a random bitstring r of length λ.
- The challenger gives $H(b||r)$ to the adversary.
- The adversary gives a bit b' to the challenger.
- The challenger returns 1 if $b = b'$ and 0 otherwise.

We then define that the adversary wins this game if the challenger returns 1 in the end.

Obviously the job of the adversary is to win whatever game it is being presented. Sometimes (such as here) a certain probability of winning is considered to be trivially achievable ($\frac{1}{2}$ by simpling making a random guess) in which case we demand that attackers have a success probability that is non-negligibly different from that value. We call the absolute value of this difference the *adversarial advantage* or just advantage and say that a scheme satisfies a notion X, if there is no attacker that has both a non negligible advantage **and** satisfies the restrictions placed upon attackers.

To justify the reason why we define the adversarial advantage as the absolute value, assume that there is an adversary $\mathcal{A}$ for which the success probability is non negligibly below the trivial value. Then there exists also an adversary $\mathcal{A}'$ that outputs the inverse of $\mathcal{A}$, meaning that the existence of one implies the existence of the other. Also note that the advantage can only be positive in the first place if the trivial success rate is 0, as there are no negative probabilities.

The other point about restrictions on the adversary is also very important: Most of the time an exponential-time adversary can break our schemes with high probability, it is just that we don't necessarily consider them to be valid. Similarly we usually don't consider adversaries with quantum computers to be valid when talking about pre quantum crypto. That doesn't mean that they don't exist though and they should be considered in practice.

(A small ungraded exercise for you: Think about whether the above security property is in any correlation to any of those that you already know from the lecture or is in fact independent.)

Receiver Indistinguishability

Let $\mathbb{G} := \langle g \rangle$ be a cyclic group of prime order q for which the DDH assumption holds (i.e. that breaking DDH is hard).

Let ElGamal refer to the ElGamal encryption scheme with $\mathbb{G}$ as group as presented in the lecture.

ElGamal has countless properties that go beyond it being just a secure encryption scheme (for a certain definition of secure). The one we are going to prove here, is receiver indistinguishability (RI). Intuitively it means that an attacker should be unable to learn any information about the intended receiver just from looking at the ciphertext, *even if he himself chooses the plaintext that is encrypted.*

This property is not self-evident: Consider any version of RSA with a keylength of λ bits and let n_0 and n_1 be different public keys with $\frac{n_1-n_0}{n_1}$ being non negligible. Then there is a non negligible chance that encrypting a random value under n_1 results in a value c such that $c > n_0$. This means however, that c is not a ciphertext under n_0, thus even a passive attacker can learn information about who the receiver might be.

There are ways around this even with RSA, but ElGamal with a standardised group that is shared by all users has that kind of protection from the very start. Intuitively this is because the first element of an ElGamal ciphertext is truly random, and the second one is the product of a known plaintext and something that is indistinguishable from randomness, meaning that from an attacker's view, given the security argument behind one-time pads, the second element is essentially random as well. But two random elements are obviously not related to any public-key, thus ElGamal offers receiver indistinguishability.

Of course the above explanation is not a formal security argument and the above „definition" of receiver indistinguishability also leaves a lot of precision to be desired; I provide them here not as an example for how to write a proof, but as an aid for understanding what is going on in the actual proof.

For our purposes we define the receiver-indistinguishability game as follows:

- The challenger samples a random bit b.
- The challenger generates to keypairs $(\mathsf{pk}_0,\ \mathsf{sk}_0)$ and $(\mathsf{pk}_1,\ \mathsf{sk}_1)$.
- The challenger gives pk_0 and pk_1 to the adversary.
- The adversary gives a plaintext m to the challenger.
- The challenger encrypts m with the public-key pk_b, giving the ciphertext c.
- The challenger gives c to the adversary.
- The adversary gives a bit b' to the challenger.
- The challenger returns 1 if $b = b'$ and 0 otherwise.

We say that the adversary wins the game if the challenger outputs 1.

We say that a scheme offers receiver indistinguishability if there is no probabilistic polynomial-time (non quantum) adversary that wins with a probability that is non negligibly different than $\frac{1}{2}$.

Much more formal:

$$\forall \mathcal{A} \in \mathrm{PPT}, \lambda \in \mathbb{N} : \left| \Pr \left[b = b' \;\middle|\; \begin{array}{l} b \xleftarrow{\$} \mathbb{B} \\ \mathsf{pk}_0,\ \mathsf{sk}_0 := \mathsf{Gen}(1^\lambda) \\ \mathsf{pk}_1,\ \mathsf{sk}_1 := \mathsf{Gen}(1^\lambda) \\ m,\ s := \mathcal{A}(\mathsf{pk}_0, \mathsf{pk}_1) \\ c := \mathsf{Enc}(\mathsf{pk}_b, m) \\ b' := \mathcal{A}(s, c) \end{array} \right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda)$$

Here PPT means the set of all probabilistic polynomial-time Turing machines, s refers to arbitrary state that the adversary $\mathcal{A}$ is allowed to retain between its invocations and negl is a negligible function.

DDH (repetition)

For the purposes of this text we use the following definition of DDH:

- The challenger samples a random bit b.
- The challenger samples random values x, y, z from $\mathbb{Z}_q$.
- If $b = 0$, the challenger gives $g^x,\ g^y,\ g^z$ to the adversary, otherwise $g^x,\ g^y,\ g^{xy}$.
- The adversary gives a bit b' to the challenger.
- The challenger returns 1 if $b = b'$ and 0 otherwise.

We say that the adversary wins the game if the challenger outputs 1.

We say that the DDH assumption holds in $\mathbb{G}$, if there is no probabilistic polynomial time (non quantum) adversary that wins with a probability that is non negligibly different from $\frac{1}{2}$.

Much more formal:

$$\forall \mathcal{A} \in \text{PPT}, \lambda \in \mathbb{N} : \left| \Pr \left[b = b' \;\middle|\; \begin{array}{l} b \xleftarrow{\$} \mathbb{B} \\ x,\ y,\ z_0 \xleftarrow{\$} \mathbb{Z}_q \\ z_1 := x \cdot y \\ b' := \mathcal{A}(g^x, g^y, g^{z_b}) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}(\lambda)$$

The actual reduction (finally!)

Using this, we can now finally look at the actual reduction. To prove that ElGamal offers receiver indistinguishability under the DDH assumption, we need to prove that we can turn every attacker against the receiver indistinguishability into an attacker against the DDH assumption.

The important part here is that we **must use the interfaces as provided**. To perform the reduction we need to take on the role of a translator between these two problems. In the one direction we need to take the role of an DDH adversary that interfaces with a DDH challenger. In the other direction we need to take on the role of a receiver-indistinguishability challenger.

Our approach will now work as follows (see also figure 1):

- We receive $g^x,\ g^y,\ g^z$ from the DDH challenger.
- We sample a bit b at random.

- We sample an exponent r from $\mathbb{Z}_q$.
- If $b = 0$:
  - We define $U := g^x$ and $V := g^r$.

  Otherwise ($b = 1$):
  - We define $U := g^r$ and $V := g^x$.
- We give U and V to the RI attacker.
- The RI attacker gives us a message m.
- We give $g^y,\ g^z \cdot m$ to the RI attacker
- The RI attacker gives us a bit b'.
- If $b = b'$, we give 1 to the DDH challenger, otherwise 0.

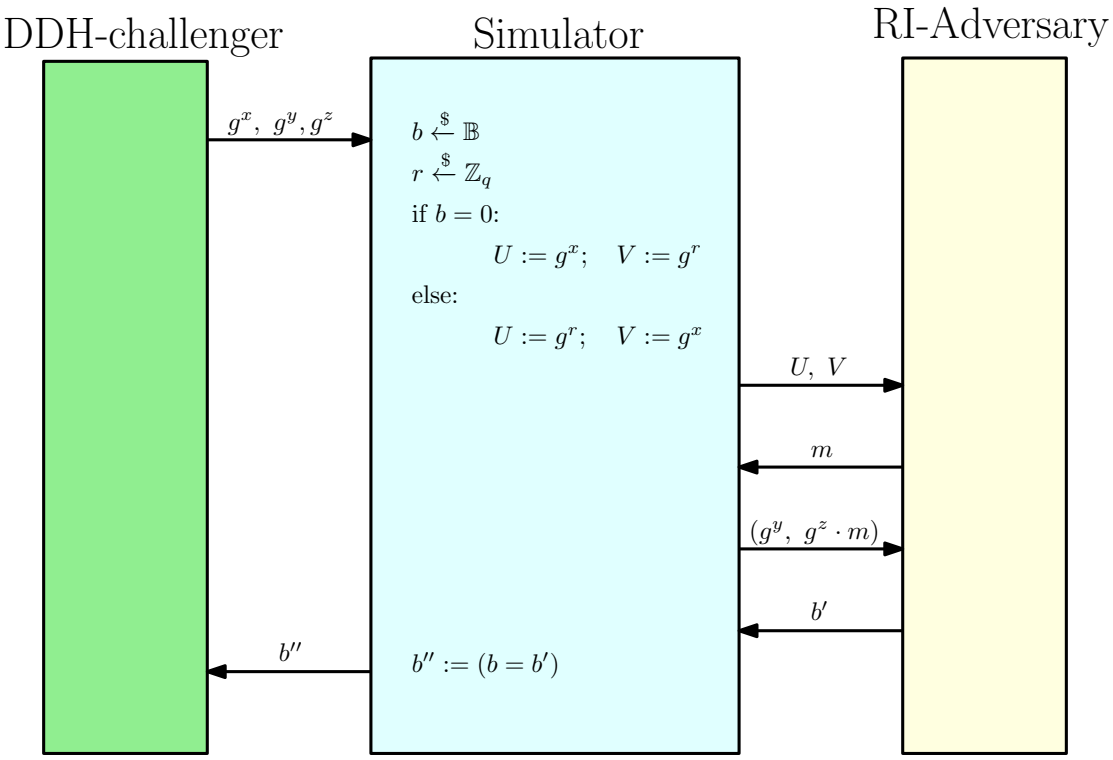

Figure 1: Graphical representation of our translator, or simulator as they are more often called.

Note that we always gave the exact kind of input to all parties that they were expecting. An important trick for this was, that we sampled the second public-key ourselves: Since we were only provided one useful value by the DDH challenger (as we need the other two values later), but the RI adversary needed two, we had to generate that value ourselves. This is an important technique that is applicable very often.

We now need to analyze why our replacement works. For this we have to consider two cases: The one in which $x \cdot y = z$ and the one in which it does not.

In the first case we simulate the RI game perfectly (jargon for: indistinguishable even for adversaries with infinite computational power): U and V are both random group elements that are sampled from the same distribution as they would be if they were both generated using the key-generation algorithm. Thus the inputs follow the expected distribution.

Since y is random by the definition of the DDH game and $z = x \cdot y$ by the assumption of this case, $(g^y,\ g^z \cdot m)$ is also perfectly indistinguishable from an honestly generated ElGamal-encryption of m. Given that we assigned the public keys so that g^x is in a random position (first or second), it is furthermore encrypted under a randomly selected public-key. (Again: The keys themselves are drawn from the same distribution so the order in which we arrange them doesn't matter.) This means that the second message also follows the expected distribution.

Note that the randomization of the position of the actual public-key is **important**: If you wouldn't do it, adversaries could win by always outputting 0 or 1 respectively. In other context different things could happen. Imagine for example that you have to break a property of a certain value but the adversary wants multiple such values. If you don't put the value into a random position, the adversary might always break the first/second/third/... value, but never the one that you want it to break.

Finally the RI adversary will output a bit b'. Let the probability that this bit is equal to b, be $\frac{1}{2} + \epsilon$. Note now that by the definition of this case, we win in the DDH game, if we output 1 to the DDH challenger. This is the case if $b = b'$. Since that is the case with probability $\frac{1}{2} + \epsilon$ we win with the same probability in this case.

Otherwise, if z was sampled at random, the situation changes: The first message that we send to the RI challenger is still indistinguishable from a real one for the very same reason as before. Now the second part of the ciphertext is however the result of multiplying m with a truly random and independent group element, resulting in a truly random group element over all. This means that it is not an encryption of m under either key. (Technically it could be, but the probability for each key is $\frac{1}{q}$, which is negligible and furthermore cancels out.) This however means that the ciphertextas a whole is in no correlation to either of the public-keys. Given that whatever bit b' the RI adversary outputs, it is independent of b. Since b is truly random, this means that the event $b = b'$ is random as well.

This means that in this case we end up outputting 0 and 1 with both probability $\frac{1}{2}$ to the DDH challenger. Since we, by definition of the case, only win the DDH game if we give 0 to the DDH challenger, this means that we win the DDH game with probability $\frac{1}{2}$ in this case.

Since either of these cases happens with probability $\frac{1}{2}$, we can compute our overall probability of winning the DDH game by simply computing the average of our success probabilities:

$$\frac{1}{2}\left(\left(\frac{1}{2}+\epsilon\right)+\frac{1}{2}\right)=\frac{1}{2}+\frac{\epsilon}{2}$$

Thus: If there is an attacker that can break the receiver-indistinguishability of ElGamal with probability $\frac{1}{2}+\epsilon$, then there is also an attacker against the DDH assumption with success probability $\frac{1}{2}+\frac{\epsilon}{2}$.

However: If ϵ is non negligible, than neither is $\frac{\epsilon}{2}$. Furthermore: If the RI adversary runs in classical probabilistic polynomial time, then so does our derived DDH attacker (technically we have to prove this, but most of the time (including here), it is so obvious that in practice nobody really cares if you just write that it is obvious).

Therefore: If there is an efficient attacker with high success probability against the receiver-indistinguishability of ElGamal (statement A), then there is also an efficient attacker with high success probability against the DDH assumption (statement B).

AKA: $A \implies B$

This implies $\neg B \implies \neg A$. In other words:

If there is no efficient attacker [...] against the DDH assumption, then there is also no efficient attacker [...] against the receiver-indistinguishability of ElGamal.

Therefore: The DDH assumption implies the receiver-indistinguishability of ElGamal.

We also say that the problem of breaking the DDH assumption can be reduced to the problem of breaking the receiver-indistinguishability, since breaking the later implies breaking both of them.

Finally, to be clear: This proof is *way* longer what I would expect of you. The reasons for this are manifold. For one it uses more text and less formulas than would be normal, in the hope that this eases understanding. Furthermore there are many more explanations on the structure and details in here than would be normal for something like this. Exercise 4b can absolutely be completed with a handful of handwritten lines.

Important Note: The terminology around these final conclusions is very confusing and often used in sloppy and strictly speaking wrong ways. As such it is easy to make mistakes here. An example for that is my correction of exercise 4c. I screwed this up in at least one case and suspect many more. If I did not award you points there with a red note that your order was wrong and you think after reading this that it is actually right, please come into my office to have it corrected. (If you got a point there despite it being wrong, consider yourself lucky, I have no way of checking this now.)

A note on my corrections

First of all, see the important note above.

Regarding the colors:

- Green means that something is correct
- Orange means that something is wrong or your wording is very bad, but I didn't subtract points for it (large scale strikeouts mean that I ignored something you wrote for your own benefit). I do explicitly not guarantee that the same mistake on a later exercise might not cause point deduction.
- Red means wrong with point deduction.
- (Rarely and only if it is unambiguous that it is from me (for example on printed papers without handwriting of your own on them): Blue are general comments on things that are neither good nor bad.)

There *are* exceptions to this, mostly if I decide later to change the grading scale, but I try very hard to ensure that there will always be some form of actual red if I deduct points.

On the third exercise sheet I also (mostly) followed the following pattern for final points per exercise:

- If there is a green hook and just a green number, this means that you got full points for the exercise.
- If there is a green hook and a fraction, the numerator (which may itself be a fraction) contains the points you received, and the denominator the maximum points.
- An underlined red zero means that you did not get any points.

The grading scale specifically was 1 point per exercise, except for exercise 2 and 4b which both got 2.

Lastly: I was serious with subtracting one point for missing staples or student IDs on the first sheet. The reason I removed those subtractions was that Tanja forbade it, not that I think it is unjustified. So please do both. (If you used a different method of connecting your sheets and I didn't write anything, I considered that method good enough as well.)