Cryptography and Security — Final Exam
Solution

Serge Vaudenay

14.1.2016

- duration: 3h
- no documents allowed, except one 2-sided sheet of handwritten notes
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade

The exam grade follows a linear scale in which each question has the same weight.

1 Attack on DL-Based Signature Schemes

In what follows, we consider a cyclic group of order q generated by some element g. We let $\langle g\rangle$ denote this group and we take multiplicative notations. We let 1 denote the neutral element. We assume that comparing and multiplying two group elements is easy and that inverting an element is easy. We assume that the discrete logarithm problem is hard in this group. In particular, we assume that $q > 2^{160}$. We further assume that we have a hash function G mapping an arbitrary group element to a $\mathbf{Z}_q$ element and a hash function H mapping an arbitrary bitstring to a $\mathbf{Z}_q$ element.

Q.1 We consider a digital signature scheme (inspired by DSA) in which the key generation and the signature algorithm work as follows:

Key generation:
1: pick $x \in \mathbf{Z}_q$ with uniform distribution
2: compute $y = g^x$
3: set the secret key to x and the public key to y

Sign m **using key** x:
1: pick $k \in \{1, 2, \ldots, 2^{128}\}$ with uniform distribution
2: compute $r = G(g^k)$
3: compute $s = \frac{H(m)+xr}{k} \bmod q$
4: set the signature to (r, s)

Verify signature (r, s) **for** m **using key** y:
1: check that $G\left(g^{\frac{H(m)}{s} \bmod q} y^{\frac{r}{s} \bmod q}\right) = r$

Prove that under a honest execution, a signature is always correct.

The difference with DSA is that r is hashed using G instead of being reduced modulo q and that k is small.
We have

$$g^{\frac{H(m)}{s} \bmod q} y^{\frac{r}{s} \bmod q} = g^k$$

As $G(g^k) = r$, the verification succeeds.

Q.2 Assume that an adversary collects many signed messages (m_i, r_i, s_i) for $i = 1, 2, \ldots, n$. If $r_i = r_j$ for $i < j$, show that the adversary can easily make a key recovery attack. How large must n be for this to happen?

HINT: first prove by an informal probability estimate that $r_i = r_j$ is most likely due to $k_i = k_j$.

As G hashes on a domain of size q and k is selected on a domain of size 2^{128}, and $q \gg 2^{128}$, collisions on k are more probable than collisions on G. So, we assume that $r_i = r_j$ is due to $k_i = k_j$.
Note that we could be a bit more precise using the Bayes formula:

$$\Pr[k_i = k_j | r_i = r_j] = \frac{\Pr[k_i = k_j]}{\Pr[r_i = r_j]} = \frac{\Pr[k_i = k_j]}{\Pr[k_i = k_j] + \Pr[r_i = r_j | k_i \neq k_j] \Pr[k_i \neq k_j]}$$

As $\Pr[k_i = k_j] \approx 2^{-128}$ and $\Pr[r_i = r_j | k_i \neq k_j] \approx \frac{1}{q}$, we obtain

$$\Pr[k_i = k_j | r_i = r_j] \approx \frac{1}{1 + \frac{1}{q}(2^{128} - 1)} \approx 1$$

If $k_i = k_j$ happens, then $s_i / s_j = \frac{H(m_i) + x r_i}{H(m_j) + x r_j} \bmod q$ so

$$s_i (H(m_j) + x r_j) \equiv s_j (H(m_i) + x r_i) \pmod{q}$$

in which $m_i, m_j, r_i, r_j, s_i, s_j$ are known. So, we can easily solve this equation in x:

$$x = \frac{s_j H(m_i) - s_i H(m_j)}{s_i r_j - s_j r_i} \bmod q$$

Collisions on k happen after $n \approx \sqrt{2^{128}} = 2^{64}$ due to the birthday paradox.

Q.3 To defeat the previous attack, our usual crypto apprentice designs the following signature scheme:

Key generation:
1: pick $x_1 \in \mathbf{Z}_q$ with uniform distribution
2: pick $x_2 \in \mathbf{Z}_q$ with uniform distribution
3: compute $y_1 = g^{x_1}$ and $y_2 = g^{x_2}$
4: set the secret key to (x_1, x_2) and the public key to (y_1, y_2)

Sign m **using key** (x_1, x_2):
1: pick $k \in \{1, 2, \ldots, 2^{128}\}$ with uniform distribution
2: compute $r_1 = G_1(g^k)$ and $r_2 = G_2(g^k)$
3: compute $s = \frac{H(m) + x_1 r_1 + x_2 r_2}{k} \bmod q$
4: set the signature to (r_1, r_2, s)

where we now use two independent hash functions G_1 and G_2 to hash group elements onto $\mathbf{Z}_q$.

Propose a verification algorithm and prove that it works.

We have

$$g^{\frac{H(m)}{s} \bmod q} y_1^{\frac{r_1}{s} \bmod q} y_2^{\frac{r_2}{s} \bmod q} = g^k$$

so we can propose to verify

$$G_i\left(g^{\frac{H(m)}{s} \bmod q} y_1^{\frac{r_1}{s} \bmod q} y_2^{\frac{r_2}{s} \bmod q}\right) = r_i$$

for $i = 1$ and $i = 2$.

Q.4 The idea of the crypto apprentice is that to adapt the attack of Q.2 to this new scheme, one needs to find i, j, ℓ such that $i < j < \ell$ and $k_i = k_j = k_\ell$. With appropriate approximations, prove that we need $n \approx 2^{86}$ to have good chances of such i, j, ℓ to exist and conclude that this attack has a too high complexity.
HINT: approximate $\log \Pr[\text{no 3-collision}]$.

Given i, j, ℓ fixed, the probability that $k_i = k_j = k_\ell$ is $1/N^2$ with $N = 2^{128}$. We have $\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$ such triplets. So, by taking $n = (2N)^{\frac{2}{3}} = 2^{86}$, we should obtain a 3-collision with good probability. More precisely, the probability should be

$$p \approx 1 - \left(1 - \frac{1}{N^2}\right)^{\frac{n(n-1)(n-2)}{6}} \approx 1 - e^{-\frac{n^3}{6N^2}} = 1 - e^{-\frac{2}{3}} \approx 49\%$$

Such n is indeed too large to be realistic.

Q.5 Ignore the idea with 3-collisions and prove that <u>two</u> regular <u>2-collisions</u> would suffice to break the new scheme. Say how large n should be for this better attack to work.
NOTE: we do <u>not</u> require a formula to give x_1 and x_2.

Given one collision $k_i = k_j$, the values of r_1 and r_2 are the same for the two signatures. We deduce the common value $x_1 r_1 + x_2 r_2 \bmod q$ with known r_1 and r_2 coming from this collision. Given a second collision, we obtain another value $x_1 r'_1 + x_2 r'_2 \bmod q$ with know r'_1 and r'_2. Hence, we can solve these two linear equations in x_1 and x_2.
We need two collisions. For that, we only need to take n a bit larger than $\sqrt{N}$. Indeed, the probability to have 2 collisions or more is

$$p \approx 1 - \left(1 - \frac{1}{N}\right)^{\frac{n(n-1)}{2}} - \frac{n(n-1)}{2} \frac{1}{N} \left(1 - \frac{1}{N}\right)^{\frac{n(n-1)}{2} - 1} \approx 1 - \left(1 + \frac{n^2}{2N}\right) e^{-\frac{n^2}{2N}}$$

So, with $n = 2\sqrt{N}$, we obtain $p \approx 59\%$. Hence $n = 2^{65}$ suffices to break the scheme.

Q.6 Upset, the crypto apprentice decides to avoid collisions by using a counter in the following scheme:

Key generation:
1: pick $x \in \mathbf{Z}_q$ with uniform distribution
2: compute $y = g^x$
3: set the secret key to x and the public key to y

4: set the counter k to a random number
5: set the e register to g^k

Sign m **using key** x:
1: increment the counter k
2: set e to eg
3: compute $r = G(e)$
4: compute $s = \frac{H(m)+xr}{k} \bmod q$
5: set the signature to (r, s)

Design a key-recovery attack for this scheme using two signatures.

In the new scheme, the signer has no exponential to compute. The value of the register e is always $e = g^k$.
Two consecutive signatures (r, s) and (r', s') on messages m and m' are computed by $r = G(g^k)$, $r' = G(g^{k+1})$, $s = \frac{H(m)+xr}{k} \bmod q$, and $s' = \frac{H(m')+xr'}{k+1} \bmod q$. Hence, we have

$$\begin{pmatrix} r & -s \\ r' & -s' \end{pmatrix} \times \begin{pmatrix} x \\ k \end{pmatrix} = \begin{pmatrix} -H(m) \\ -H(m') + s' \end{pmatrix} \pmod q$$

and we deduce

$$x = \frac{ss' - sH(m') + s'H(m)}{sr' - s'r} \bmod q$$

Q.7 What if we now use the following scheme?

Key generation:
1: pick $x \in \mathbf{Z}_q$ with uniform distribution
2: compute $y = g^x$
3: set the secret key to x and the public key to y
4: set the counter k to a random number
5: pick $\mathsf{inc} \in \mathbf{Z}_q^*$ with uniform distribution
6: set the e register to g^k
7: set the e' register to g^{inc}

Sign m **using key** x:
1: set k to $k + \mathsf{inc}$
2: set e to ee'
3: compute $r = G(e)$
4: compute $s = \frac{H(m)+xr}{k} \bmod q$
5: set the signature to (r, s)

Again, we always have $e = g^k$, and the values of k and e are updated consistently by minimizing the cost for the signer. We have $k = k_0 + i \times \mathsf{inc}$, where k_0 is the initial value of k.
With 3 consecutive signatures (m_i, r_i, s_i) for $i = 1, 2, 3$, we have equations of form

$$s_1 k_1 = H(m_1) + x r_1$$
$$s_2(k_1 + \mathsf{inc}) = H(m_2) + x r_2$$
$$s_3(k_1 + 2\mathsf{inc}) = H(m_3) + x r_3$$

modulo q, where the unknowns are k_1, inc, and x. So, this is a linear system which can be easily solved.

2 RSA in an Extension Ring

Q.1 Let p be a prime number such that $p \bmod 4 = 3$. We consider the polynomial $x^2 + 1$ in the ring $\mathbf{Z}_p[x]$ of polynomials in the indeterminate x, with coefficients in $\mathbf{Z}_p$. Prove that $x^2 + 1$ is irreducible.

Since $p \bmod 4 = 3$, we have $(-1)^{\frac{p-1}{2}} = (-1)$ so -1 is not a quadratic residue in $\mathbf{Z}_p$. Hence, x^2+1 has no root in $\mathbf{Z}_p$. As it is of degree 2, this implies that it is irreducible in $\mathbf{Z}_p[x]$. (Otherwise, we would reduce it into some $x^2 + 1 = (ax + b)(cx + d)$ with a and c nonzero and we would obtain the roots $-b/a$ and $-d/c$.)

Q.2 Let p be a prime number such that $p \bmod 4 = 3$. We consider the set $K = \mathbf{Z}_p[x]/(x^2 + 1)$ of all polynomials over $\mathbf{Z}_p$ taken modulo $x^2 + 1$. This defines the addition and the multiplication over K. (This is just the regular addition and multiplication of polynomials reduced modulo $x^2 + 1$ and modulo p.) Give the cardinality of K and say what type of algebraic structure it has. Justify your answer.

That is actually the standard construction of the finite field $\mathsf{GF}(p^2)$ since $x^2 + 1$ is monic, irreducible, and of degree 2.
By reducing an arbitrary polynomial modulo $x^2 + 1$, we always obtain a polynomial of degree bounded by 1. It can be written $a + bx$ for two coefficients a and b. Now, no two distinct such elements can be equal modulo $x^2 + 1$: if $a + bx \equiv a' + b'x$ modulo $x^2 + 1$, it means that $(b - b')x + (a - a')$ is a multiple of $x^2 + 1$, which implies that $b - b' = 0$ and $a - a' = 0$, hence $a = a'$ and $b = b'$. So, we have exactly p^2 elements in K.
By construction, we obtain a ring. We further check that every nonzero element $a+bx$ is invertible. Indeed, the function $(c + dx) \mapsto (a + bx)(c + dx) \bmod (x^2 + 1) \bmod p$ is linear and has no nonzero preimage of 0. (Indeed, if $(a + bx)(c + dx) = 0$ modulo $x^2 + 1$, then $(a + bx)(c + dx)$ is a multiple of $x^2 + 1$. If both $a + bx$ and $c + dx$ are nonzero, due to their degree being bounded by 1 with a product of degree exactly 2 they must be of degree exactly 1, so they must be divisors of $x^2 + 1$, which is impossible. So, either $a + bx$ or $c + dx$ must be zero.) So, this linear function is a bijection of K and it has a preimage of 1 which is the inverse of $a + bx$. Therefore, K is a field of p^2 elements.
(We could have a shorter proof with more background in algebra.)

Q.3 Let p and q be two different prime numbers such that $p \bmod 4 = q \bmod 4 = 3$. Let $n = pq$. Let $R = \mathbf{Z}_n[x]/(x^2+1)$ be the set of all polynomials over $\mathbf{Z}_n$ taken modulo x^2+1. We want to construct an RSA-like cryptosystem over R.

Prove that there are exactly $\phi = (p^2-1)(q^2-1)$ invertible elements in R.

HINT: either count or think Chinese.

One way is to count the number of non-invertible elements. First, we can use the property that an element $a+bx$ is non-invertible is equivalent to the property that a and b are either both divisible by p or both divisible by q. The $\Leftarrow$ implication is trivial as a product with any candidate for the inverse would stay divisible by p or q and 1 is not. For the $\Rightarrow$ implication, we show that if among a and b there is at least one which is not divisible by p and one which is not divisible by q then $a+bx$ is invertible. For that, we first observe that a^2+b^2 is nonzero modulo p (otherwise, a/b or b/a would be a square root of -1 modulo p, which is impossible), and similarly nonzero modulo q, so it is invertible modulo n. Then, we realize that $(a-bx)/(a^2+b^2) \bmod n$ is the inverse of $a+bx$.

If N_p (resp. N_q, N_n) is the number of elements which are divisible by p (resp. q, n), by the principle of inclusion/exclusion, we have a number of invertible elements equal to $n^2-N_p-N_q+N_n$. As $N_p = q^2$, $N_q = p^2$, and $N_n = 1$, we have $n^2-p^2-q^2+1$ invertible elements, which is $(p^2-1)(q^2-1)$.

We now show the same using the Chinese remainder theorem.

As $(a+bx)+(c+dx) = (a+c)+(b+d)x$ and

$$(a+bx) \times (c+dx) \equiv (ac-bd)+(ad+bc)x \pmod{x^2+1}$$

R is isomorphic to $\mathbf{Z}_n^2$ where we define

$$(a,b)+(c,d) = (a+c, b+d)$$

and

$$(a,b) \times (c,d) = (ac-bd, ad+bc)$$

where all numbers are taken modulo n. These operations are polynomial modulo n. So, due to the Chinese Remainder Theorem, all operations over $\mathbf{Z}_n$ are equivalent to operations over $\mathbf{Z}_p \times \mathbf{Z}_q$ (note that p and q are different primes, so they are coprime). So, R is isomorphic to $\mathbf{Z}_p^2 \times \mathbf{Z}_q^2$ where the operations over $\mathbf{Z}_p^2$ and $\mathbf{Z}_q^2$ are defined as above, like in $\mathbf{Z}_n^2$. These structures are isomorphic to $\mathbf{Z}_p[x]/(x^2+1)$ and $\mathbf{Z}_q[x]/(x^2+1)$. So, we obtain that R is isomorphic to $\mathbf{Z}_p[x]/(x^2+1) \times \mathbf{Z}_q[x]/(x^2+1)$ which is a ring obtained by the product of two finite fields $\mathsf{GF}(p^2)$ and $\mathsf{GF}(q^2)$.

In a product of two fields, an element is invertible if and only if both components are nonzero. So, we have $\phi = (p^2-1)(q^2-1)$ invertible elements.

Q.4 Under the same hypothesis as in Q.3, we want to encrypt an element $m \in R$ by computing m^e and to decrypt by raising to the power d. How to set e and d for the decryption to work correctly? Justify your answer.

For $m \in R$, we want to have $m^{ed} = m$. So, $m^{ed-1} = 1$ for all $m \in R^*$. Hence, $ed-1$ must be a multiple of all element orders. One way to achieve this is to take $ed \bmod \phi = 1$. So, we can take e such that $\mathsf{gcd}(e, \phi) = 1$ and take d as the inverse of e modulo ϕ.
To be more precise, we can see from the previous question that R^* includes a subgroup isomorphic to $\mathsf{GF}(p^2)^*$, which is cyclic. (If we write $R \sim \mathsf{GF}(p^2) \times \mathsf{GF}(q^2)$, this is the subgroup of all $(t, 1)$ for $t \in \mathsf{GF}(p^2)^*$.) So, $ed-1$ must be a multiple of p^2-1. This is the same for q^2-1. So, $ed-1$ must be a multiple of $\lambda = \mathsf{lcm}(p^2-1, q^2-1)$, which is the exponent of R^*. So, we can take e such that $\mathsf{gcd}(e, \lambda) = 1$ and take d as the inverse of e modulo λ.
Once we have seen that $m^{ed} = m$ for all $m \in R^*$, we can verify it for all $m \in R$ by using the Chinese Remainder Theorem: for any polynomial m, if $m \equiv 0$ modulo x^2+1 and modulo p, then if $m^{ed} \equiv m$ as well. Otherwise, m has order multiple of $ed-1$ modulo x^2+1 and modulo p, so $m^{ed} \equiv m$ anyway. Hence, $m^{ed} \equiv m$ modulo x^2+1 and modulo p. It is the same modulo q. So, $m^{ed} \equiv m$ modulo x^2+1 and modulo n.

Q.5 In the context of Q.4, can we take $e = 3$? Justify your answer.

We must take $ed \bmod \phi = 1$ so $\mathsf{gcd}(e, \phi) = 1$.
We cannot take $p = 3$, otherwise it is trivial to factor n. So, p is coprime with 3. Hence, either $p+1$ or $p-1$ is a multiple of 3. We deduce that p^2-1 is always a multiple of 3. So, $e = 3$ is not invertible modulo ϕ (and not modulo λ either). Therefore, $e = 3$ is not possible.

Q.6 By selecting the last decimal digit of p to be equal to 7 and the last decimal digit of q to be equal to 3, prove that we can always use $e = 5$ in the previous construction.

If $p \bmod 10 = 7$ then $p \bmod 5 = 2$ so $p^2 - 1 \bmod 5 = 3$. If $q \bmod 10 = 3$ then $q \bmod 5 = 3$ so $q^2 - 1 \bmod 5 = 3$. So, $\phi \bmod 5 = 4$ so 5 does not divide ϕ. Since 5 is prime, we deduce that 5 is invertible modulo ϕ. So, we can take $e = 5$.

Q.7 Is there any advantage of this cryptosystem compared to RSA? Explain why.
HINT: compare the security with respect to modulus size, key and message lengths, and complexities.

So far, the best way to break RSA is to factor n. So, we have to take n long enough to make it hard to factor. If we can factor n, we can also break the new cryptosystem. So, in practice, the length requirements on n are the same for RSA and the new cryptosystem.
Clearly, the e exponent may have the same size but the d exponent is twice larger. The messages are twice larger as well (so we can encrypt more).
With the same modulus length, the complexity of the new scheme is larger than for RSA (say four times larger if we implement multiplication in a straightforward way). So, there seems to be no advantage in using this new cryptosystem.

3 On Securing Biometric Passports

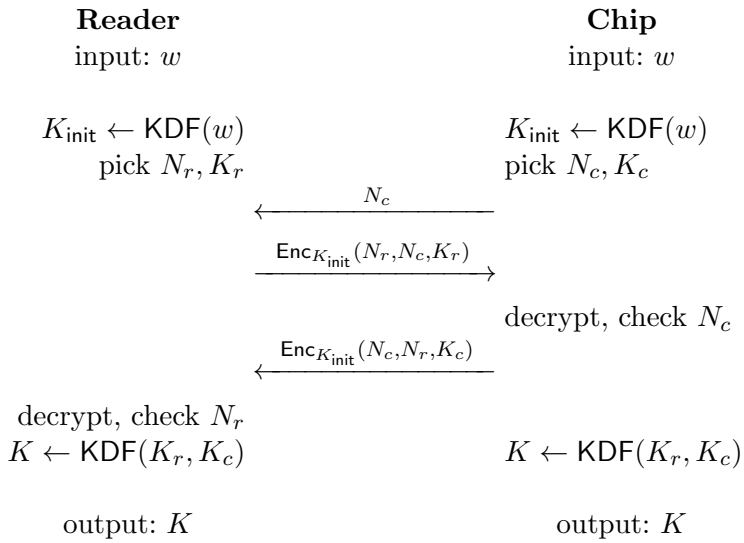
A biometric passport is an identity document with a contactless chip. Reading the digital identity works like this:

1: The reader first reads the low-entropy password w which is printed inside the passport.
2: The reader sends a standard RFID broadcast signal and the chip responds.
3: The chip requests to go through a password-based key agreement. The password w is the input of the protocol on the reader side. On the chip side, there is a long-term public/secret key pair pk/sk and w. (sk is stored in the chip but is not accessible to the reader.) At the end of the protocol, the output on both sides is a symmetric key K.
4: The reader and the chip communicate securely by using this key K.
5: Through this secure communication, the reader can retrieve some files containing the identity information ID, a biometric reference template bio, the public key pk again, and a signature σ from the issuing country that $(\mathsf{ID}, \mathsf{bio}, \mathsf{pk})$ is correct.
6: The reader extracts from ID the field $\mathsf{country}$ indicating the issuing country. It is assumed that the reader has previously got in a secure way the root certificate C_{country} from the issuing country so that he can verify σ.

Then, the reader has obtained $(\mathsf{ID}, \mathsf{bio})$ which can then be used to identify the person.

We further describe BAC, the original password-based key agreement protocol which is in the standard. In this exercise, some questions are specific to BAC.

BAC makes no use of any pk/sk pair. It works as follows: the reader and the chip derive $K_{\mathsf{init}} = \mathsf{KDF}(w)$ using a key derivation function, select some random nonces N_r (for the reader) and N_c (for the chip) and some keys K_r (for the reader) and K_c (for the chip); the chip sends N_c in clear to the reader; the reader sends (N_r, N_c, K_r) securely (using K_{init}) to the chip; the chip checks that N_c is correct and sends (N_c, N_r, K_c) securely (using K_{init}) to the reader; the reader checks that N_r is correct; the reader and the chip derive $K = \mathsf{KDF}(K_r, K_c)$.



Q.1 If the password-based key agreement protocol makes no use of any pk/sk pair like in BAC, prove that when the holder shows his biometric passport to someone (for instance, at the hotel check in counter), this person can easily copy the passport. How could this be fixed?

What the chip needs is just w, ID, bio, and σ.
Someone who has seen the passport knows w. So, he can access to the chip and read all other information. This information could be then stored in a blank biometric passport.
Then, the reader sees no difference between the original passport and the copied one as they contain the same information.
This was fixed in the original standard by having an additional "Active Authentication" protocol using the pk/sk pair using public-key cryptography. As the value of sk cannot be copied (it is not accessible), we cannot make a copy which simulates the protocol knowing sk.
But we could also replace BAC with a better password-based key agreement protocol instead of adding another protocol.

Q.2 If an adversary has obtained w (by whatever means), what is the threat for the holder of the passport? Describe a possible scenario.

With w compromised, it is then easy to trace the movements of the passport as we can easy recognize it by running the protocol again. We can also start communication with the chip and read the private data $(\mathsf{ID}, \mathsf{bio}, \sigma)$. In both cases, this is a privacy concern.
If, like in the case of Q.1, the protocol makes no use of sk, the additional threat related to having read $(\mathsf{ID}, \mathsf{bio}, \sigma)$ is that one could make a digital copy the passport. But this is not really the question here. The main threat remains the loss of privacy, regardless of the use of sk.
A possible scenario is that we obtain w from a legitimate physical access to the document (for instance, at a hotel check in desk), then, implement sensors to trace the holder carrying his passport. We can recognize it when he enters again in the hotel, or in a shop, etc.

Q.3 If we use BAC as a password-based key agreement protocol, prove that the password w and all transmitted data can be recovered in clear with a passive offline exhaustive search. How could we replace BAC to avoid this attack?

A passive adversary can get N_c (which is sent in clear) and $c = \mathsf{Enc}_{K_{\mathsf{init}}}(N_r, N_c, K_r)$. Then, he can do an exhaustive search on w (which has a low entropy) until $\mathsf{Dec}_{\mathsf{KDF(guess)}}(c)$ is of form $(., N_c, .)$.
This way, the adversary recovers w.
Even worse: the adversary can then decrypt the messages and deduce K, then decrypt further communication which transmits the private data.
The only way to fix it is to use public-key cryptography and a correct PAKE protocol which is secure against offline exhaustive search instead of BAC.

Q.4 One difference between a regular identity document (with ID and a picture bio printed) with an official stamp σ and a digital document $(\mathsf{ID}, \mathsf{bio}, \mathsf{pk})$ with a digital signature σ is that we cannot use a photocopy of the stamped document as a proof, whereas we can use an electronic copy of the digital signed document like the original one.
What is the potential threat related to this difference? Explain the related cryptographic notion and a possible scenario. How could this problem be fixed?

The digital signature σ is transferable because it can be perfectly copied: copies of a digital signature are undeniable. The important cryptographic property is that (copies of) the signature are transferable and undeniable because they are unforgeable.
The original stamped document is assumed to be unforgeable. Although it can be copied, it cannot be perfectly *copied as copies could be forged. So, photocopies are deniable.*
So, the original document is a proof while its copies are not.
The threat is for people who want to hide some sensitive part of their identity (such as the exact age, official name, citizenship, etc) as any copy with σ which is disclosed would leak evidence of the private data.
To fix this, the passport should not give the transferable σ but rather go through an interactive proof of knowledge for a valid signature. This interactive proof should be deniable (e.g. zero-knowledge).
Note: again, this question is not really about copying passports. If we use sk*, we can tell genuine passports and their copies apart. The question is about copying the signature for publication.*