Security Protocols and Application — Final Exam

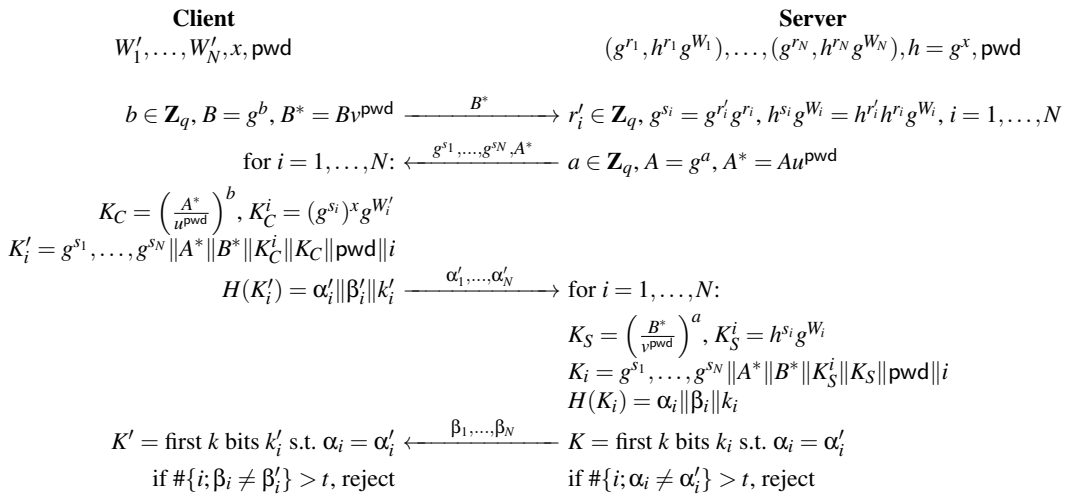
Philippe Oechslin and Serge Vaudenay

17.6.2013

- allover exam duration: 2h00
- no document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

1 Security Analysis of a Multi-Factor Authenticated Key Exchange Protocol

We recall that the MF-AKE authentication protocol works over a group of large prime order q generated by some g. We use two random group elements u and v. The protocol is depicted on the following diagram:



Q.1 What is the role of x, h, pwd, and the W_i's?

Q.2 What is the difference between W_i and W'_i?

Q.3 Why don't we store W_i instead of $(g^{r_i},h^{r_i}g^{W_i})$ on the server side?

Q.4 Explain how the server computes g^{s_i} and $h^{s_i}g^{W_i}$.

Q.5 If the protocol succeeds, explain why we have $K=K'$.

Q.6 Show that by selecting $s_i\in\mathbf{Z}_q$ at random, $i=1,\dots,N$, an adversary who knows pwd and h and tries to play the role of the server can recover the W'_i's.

2 Hash DoS attacks

AWK is simple scripting language available on all unix platforms. The language uses a hash fonction to implement arrays of strings. In the latest version of GNU awk (4.1.0) we can find the following comments in the source code:

```
/*
 * Even more speed:
 * #define HASHC   h = *s++ + 65599 * h
 * Because 65599 = pow(2, 6) + pow(2, 16) - 1 we multiply by shifts
 *
 * 4/2011: Force the results to 32 bits, to get the same
 * result on both 32- and 64-bit systems. This may be a
 * bad idea.
 */
```

The hash function iterates through all characters of a string. It multiplies the current value of the hash h with the constant 65599 and adds the next character s to the hash. Initially h is zero.

Q.1 This hash function can be subjected to an equivalent substring attack. Explain how this attack works and why this hash is vulnerable to it.

Q.2 You want to mount the equivalent substrings attack using strings made of an alphabet of 64 different characters (2^6). You want to find one pair of colliding substrings. How long must your substrings be such that you can be reasonably sure that you will find one pair of colliding substrings? Explain the assumptions that you made to find your result.

Q.3 If the maximum length of the parameter you can pass to an AWK script is 64 characters long, how many coliding strings can you construct using two colliding substrings ?

Q.4 What is the number of operations that AWK will have to carry out to insert all these strings into an array ?

Q.5 Now assume the script only accepts parameters of up to nine characters. A colliding substring attack will not be efficient and we have to try a meet-in-the-middle attack.
Which property must the hash function have to make a meet-in-the-middle attack possible ?

Q.6 Explain the operations that have to be carried out to mount a meet-in-the-middle attack against a hash function h.

Q.7 Find the parameters of the meet-in-the-middle attack that will generate 2^{21} colliding 9-character strings with the smallest possible number of operations while keeping the memory usage low. Give the number of operations and the memory needed for your attack.

Q.8 The meet-in-the-middle attack appears to be better since it can create large numbers of collisions on shorter strings than the equivalent substring attack. Assuming the hash function is vulnerable to both types of attack, can you describe a case were the equivalent substring attack has an advantage over the meet-in-the-middle attack ?

Q.9 Siphash is a 64 bit hash function that was designed to replace the hash functions that are vulnerable to hash dos attacks. Give two reasons why Siphash is superior to a cryptographic hash function like SHA1, truncated to 64 bits.