Cryptography, winter 2014/15

Identification and authentication

Dr. Daniel Loebenberger

An identification protocol must safeguard against the following misuses, when Alice identifies herself to Bob:

- If Eve intercepts the transmission, she should not be able to impersonate Alice later.
- Bob should not be able to impersonate Alice.

These requirements preclude the possibility of just Alice sending a message to Bob. Most identification schemes incorporate three steps:

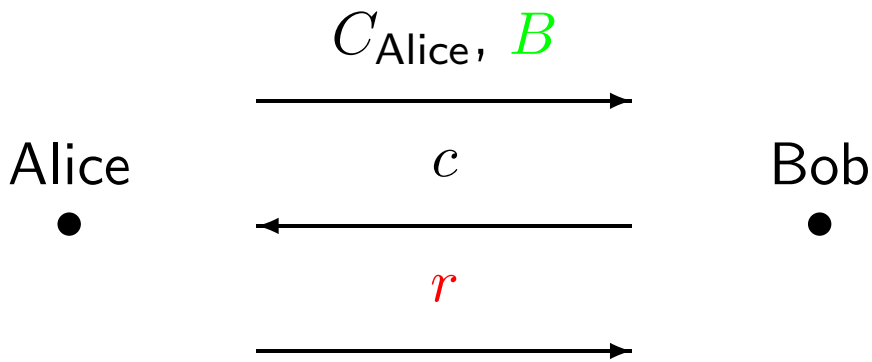
1. Alice sends information to Bob.
2. Bob issues a *challenge* to Alice.
3. Alice responds.

If Alice's response is satisfactory, then Bob will assume that it is really Alice who is talking to him.

ALGORITHM. Schnorr's identification scheme.

Input: Publicly known $G = \langle g \rangle$, d, ID(Alice), and $A = g^a$.
Known to Alice: C_{Alice}.

1. Alice chooses a secret session key $b \xleftarrow{\text{🎲}} \mathbb{Z}_d$ and sends C_{Alice} and the public version $B = g^b \in G$ to Bob.
2. Bob checks that $\text{ver}_{\text{TA}}((\text{ID(Alice)}, A), s) =$ "true", chooses $c \xleftarrow{\text{🎲}} \mathbb{Z}_d$, and sends c to Alice.
3. Alice computes $r = b + ac$ in $\mathbb{Z}_d$ and sends it to Bob.
4. Bob accepts Alice as herself if $BA^c = g^r$, and not if otherwise.


C_{Alice}, B
Alice
c
Bob
r

Theorem

The protocol works correctly, that is, if it is properly executed, then Bob will accept Alice's identification. The total computing time is $O(n^3)$ bit operations, and the number of bits communicated is

$$\text{length}(\text{ID}(\text{Alice})) + \text{length}(\text{TA signature}) + 4n.$$

Example

We arrive at the following concrete numbers. We assume that ID(Alice) is a string of 512 bits. If we use the 8-bit extended ASCII coding of $2^8 = 256$ characters, then this corresponds to $512/8 = 64$ letters. The DSA on elliptic curves provides signatures of $2 \cdot 224 = 448$ bits for the recommended bit size of 224 for group cryptography. A, B, c, and r have 224 bits each. In total, Alice's certificate comprises $512 + 224 + 448 = 1184$ bits and $1184 + 3 \cdot 224 = 1856$ bits are transmitted.

Alice's computation is the choice of b, the modular exponentiation g^b in G in step 1, and $b + ac$ in $\mathbb{Z}_d$ in step 3. The latter is mainly one multiplication modulo d, but the former are on average about $224 \cdot 1.5 = 336$ multiplications in the group G.

If this is to run on a *smart card* with very limited computing capability, then the computation of g^b is a bottleneck. But it can be performed *off-line*: after each identification, the smart card already starts computing b and g^b and has them ready to dispatch at the next identification process.

Theorem

The discrete logarithm problem DL_G can be reduced to double impersonation in Schnorr's identification system.

ALGORITHM. Okamoto's identification scheme.

Input: Publicly known $G = \langle g \rangle$, d, ID(Alice), and $A = g^{a_1} g^{a_2}$.
Known to Alice: C_{Alice}.

1. Alice chooses $b_1, b_2 \in \mathbb{Z}_d$ at random and sends her certificate $C_A = (\text{ID(Alice)}, A, s)$ and $B = g_1^{b_1} g_2^{b_2}$ to Bob.
2. Bob verifies that $\text{ver}_{\text{TA}}((\text{ID(Alice)}, A), s) =$ "true" and chooses $c \xleftarrow{\text{🎲}} \mathbb{Z}_d$.
3. Alice sends

$$r_1 = b_1 + a_1 c \quad \text{and} \quad r_2 = b_2 + a_2 c \text{ in } \mathbb{Z}_d$$

   to Bob.
4. Bob verifies that $BA^c = g_1^{r_1} g_2^{r_2}$.

Lemma

If Eve has a value B for which she can impersonate Alice for at least two values of c, then Eve can easily compute $e_1, e_2 \in \mathbb{Z}_d$ such that $A = g_1^{e_1} g_2^{e_2}$.

Theorem

If Eve has a value B for which she can impersonate Alice for at least two values of r, independently of $\mathsf{sk}_{\mathsf{Alice}}$, then Alice and Eve together can, with probability at least $1 - d^{-1}$, easily compute $k = \mathsf{dlog}_{g_2} g_1$.