

EIDMA-Stieltjesweek Graduate Course

Σ -protocols

September 23, 2003

Berry Schoenmakers
TU Eindhoven

`berry@win.tue.nl`

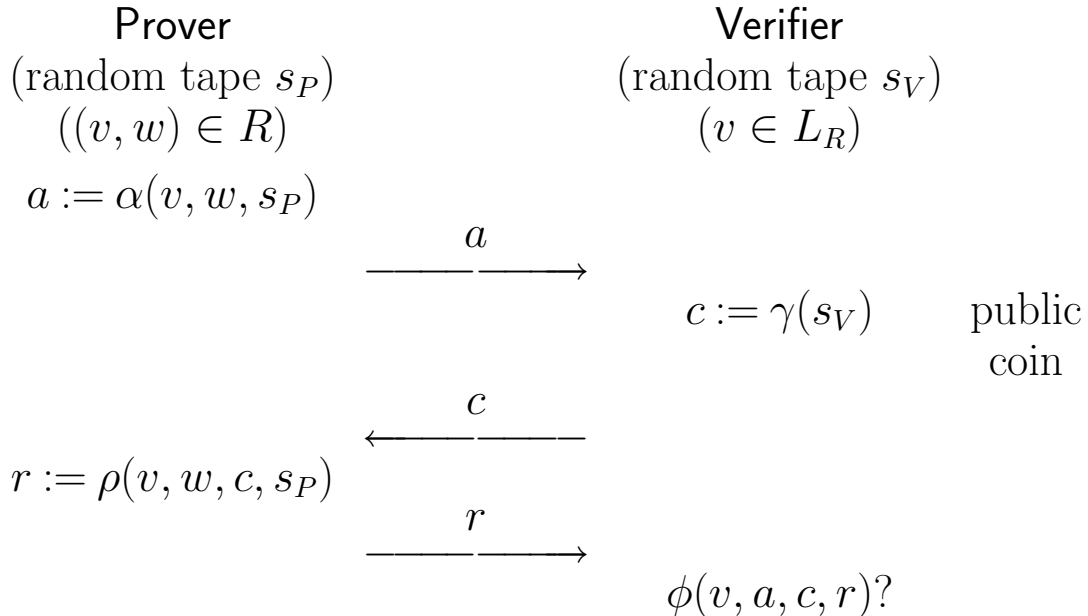
<http://www.win.tue.nl/~berry/>

1. Definitions

1.1. Σ -protocols

Let $R = \{(v, w)\}$ be a binary relation. (It is assumed that for some given polynomial p that $|w| \leq p(|v|)$ for all $(v, w) \in R$.) Here, v denotes the common input to prover and verifier, and w denotes a witness, which is the private input to the prover. Let $L_R = \{v \mid \exists w : (v, w) \in R\}$.

A Σ -protocol for relation R is of the following form:



1.2. Security properties for Σ -protocols

Completeness: if P and V follow the protocol, the verifier always accepts.

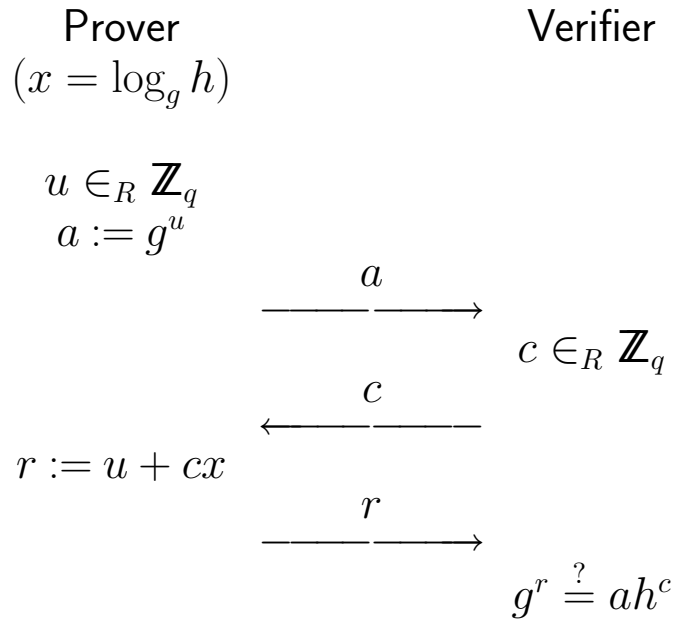
Special soundness: for any v and any pair of accepting conversations (a, c, r) and (a, c', r') with $c \neq c'$ one can efficiently compute witness w such that $(v, w) \in R$.

Special honest-verifier zero-knowledge: there exists a p.p.t. machine S (simulator) which for any v and c produces conversations (a, c, r) with the same probability distribution as conversations between the honest P and V with common input v and challenge c .

Note that a cheating prover succeeds with probability at most $1/q$, where q denotes the cardinality of the challenge space $\gamma(\cdot)$.

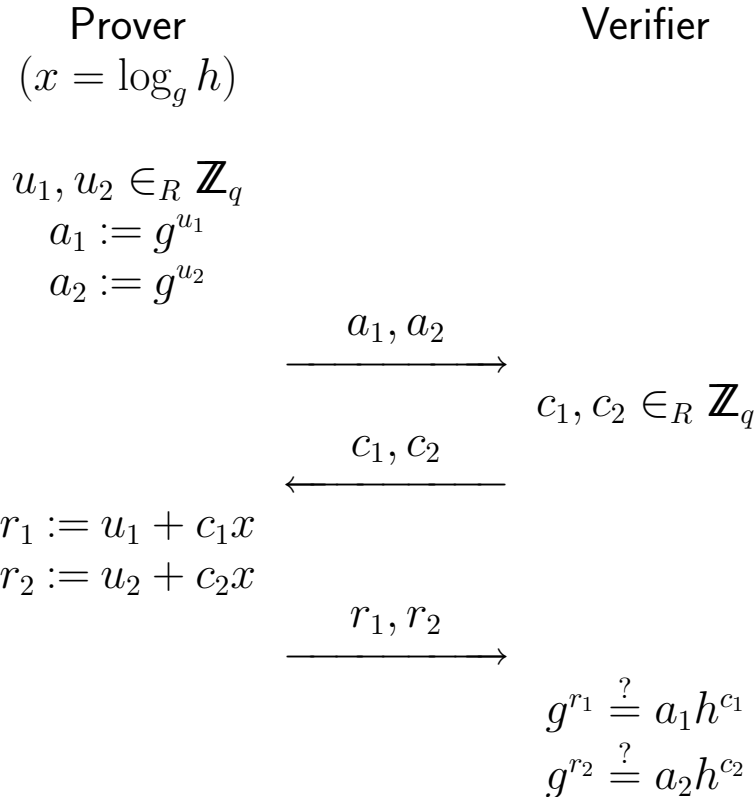
2. Schnorr-based examples

2.1. Schnorr's protocol



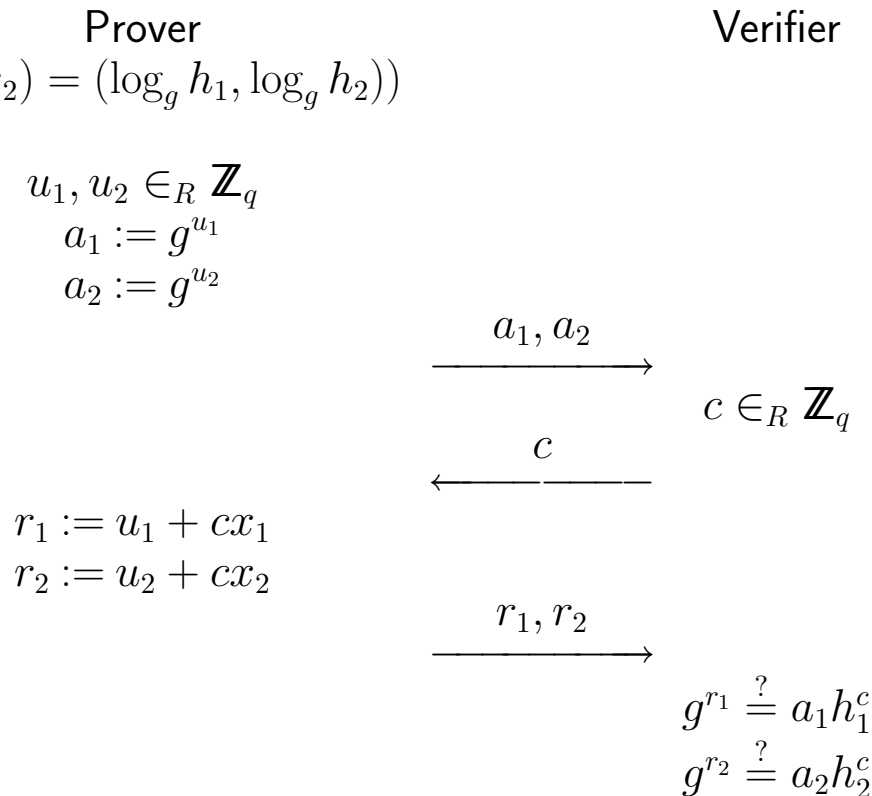
2.2. Parallel composition

Running two instances of Schnorr's protocol in parallel, for the *same* public key h , results in a Σ -protocol with a larger challenge length.



2.3. AND composition

Given two public keys h_1, h_2 , one proves knowledge of $\log_g h_1$ and $\log_g h_2$, by running two instances of the Schnorr proof in parallel, using a *common* challenge.



2.4. OR composition

It turns out that there is a proof of knowledge of (at least) one of $x_1 = \log_g h_1$ and $x_2 = \log_g h_2$ of the *same* complexity as an AND proof.

We let the prover do a proof of knowledge for both $\log_g h_1$ and $\log_g h_2$ in parallel but giving the prover one degree of freedom in choosing the two challenges for these proofs. This allows the prover to *cheat* in one of the two proofs.

Suppose the prover knows x_1 but does not know x_2 . The prover will then do a real proof of knowledge for $\log_g h_1$, and use the honest-verifier zero-knowledge property of the Schnorr protocol to create a simulated proof for $\log_g h_2$.

Prover

Verifier

(using $x_2 = \log_g h_2$) | (using $x_1 = \log_g h_1$)

$$\begin{aligned} r_1, c_1, u_2 &\in_R \mathbb{Z}_q \\ a_1 &:= g^{r_1} h_1^{-c_1} \\ a_2 &:= g^{u_2} \end{aligned}$$

$$\begin{aligned} r_2, c_2, u_1 &\in_R \mathbb{Z}_q \\ a_1 &:= g^{u_1} \\ a_2 &:= g^{r_2} h_2^{-c_2} \end{aligned}$$

$$\xrightarrow{a_1, a_2}$$

$$\xleftarrow{c}$$

$$c \in_R \mathbb{Z}_q$$

$$\begin{aligned} c_2 &:= c - c_1 \\ r_2 &:= u_2 + c_2 x_2 \end{aligned}$$

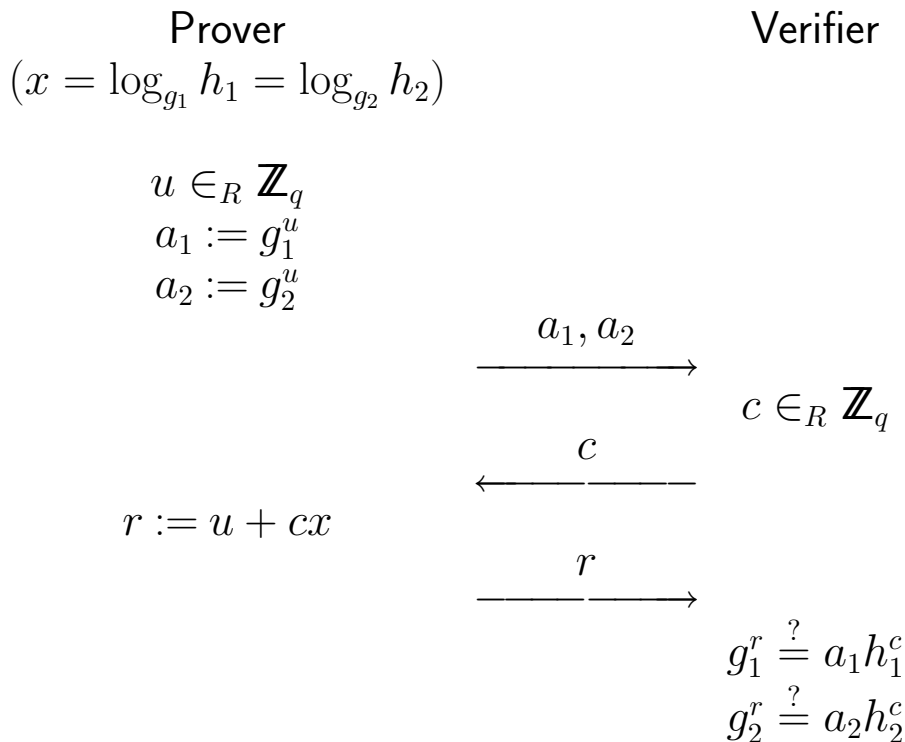
$$\begin{aligned} c_1 &:= c - c_2 \\ r_1 &:= u_1 + c_1 x_1 \end{aligned}$$

$$\xrightarrow{c_1, c_2, r_1, r_2}$$

$$\begin{aligned} c_1 + c_2 &\stackrel{?}{=} c \\ g^{r_1} &\stackrel{?}{=} a_1 h_1^{c_1} \\ g^{r_2} &\stackrel{?}{=} a_2 h_2^{c_2} \end{aligned}$$

2.5. Equality of Discrete Logs

Given two public keys $h_1 = g_1^x, h_2 = g_2^x$, one proves knowledge of $x = \log_{g_1} h_1 = \log_{g_2} h_2$, by running two instances of the Schnorr proof in parallel, using a *common* random choice, a *common* challenge and a *common* response.



2.6. Schnorr signatures

Schnorr signatures are obtained by applying the Fiat-Shamir heuristic to Schnorr's protocol: compute the challenge as a hash $\mathcal{H}(\cdot)$ of the message m and the value a .

Signer	Receiver
$(x = \log_g h)$	
$u \in_R \mathbb{Z}_q$	
$a := g^u$	
$c := \mathcal{H}(m, a)$	
$r := u + xc$	$\xrightarrow{a, r} c := \mathcal{H}(m, a)$
	$g^r \stackrel{?}{=} ah^c$

(As an optimization, one may send c instead of a , as the bit-length of c may be much smaller than the bit-length of a . The receiver computes $a := g^r h^{-c}$ and accepts if $c = \mathcal{H}(m, a)$.)

The Fiat-Shamir technique for converting Σ -protocols into signature schemes is provably secure in the so-called random oracle model.

3. Exercises

Exercise 1 *Prove the special soundness of the OR composition for the Schnorr protocol.*

Exercise 2 *Let g, h denote generators of a group G of large prime order q such that $\log_g h$ is unknown to anyone. Let $B = g^x h^y$ denote the common input to prover and verifier, where $x, y \in \mathbb{Z}_q$ is private input to the prover. For each of the following predicates $\psi(x, y)$, design a Σ -protocol that proves knowledge of $x, y \in \mathbb{Z}_q$ such that $B = g^x h^y$ and $\psi(x, y)$ holds:*

a. $\psi(x, y) \equiv \text{true};$

b. $\psi(x, y) \equiv x = y;$

c. $\psi(x, y) \equiv \alpha x + \beta y = \gamma$ for given $\alpha, \beta, \gamma \in \mathbb{Z}_q;$

d. $\psi(x, y) \equiv x \in \{0, 1\};$

e. $\psi(x, y) \equiv x \in \{0, \dots, 2^k - 1\}$, where k is a fixed integer, $1 \leq k \leq \lfloor \log_2 q \rfloor;$

f. $\psi(x, y) \equiv x \neq 0;$

g. $\psi(x, y) \equiv \exists a \in \mathbb{Z}_q : x = a^2;$