Three-Move Identification Schemes

Helger Lipmaa

Küberneetika AS; http://www.cyber.ee

helger@cyber.ee; http://home.cyber.ee/helger

April 23, 1999

Introduction

The first known three-move protocol is by Fiat and Shamir (based on the difficulty of factoring), extended later by Fiat, Feige and Shamir (1988) and finally by Feige and Shamir (1990) that defined the notion of "witness hiding". Other desirable objectives are: special HVZK, collision intractability, proofs of knowledge, special soundness. A witness hiding proof of knowledge can be used as a secure identification scheme.

More identification schemes of the same type:
1988 — Guillou-Quisquater
1990 — Ong-Schnorr (witness hiding, !PoK, CI, !SS)
1991 — Schnorr (SS, SHVZK, PoK)
1992 — Brickell-McCurley (witness hiding)
1992 — Okamoto-Schnorr (witness hiding)

Notation

If α is an algorithm, then the notation "$a \leftarrow \alpha(b)$" refers to the computation of the output "a", on input bit string "b". If V is a set, $v \leftarrow V$ denotes uniform and random selection of an element v from V.

Red variables are known only to A. Blue variables are known only to B.

Motivation, I

RSA based authentication, where w (*witness*) is the secret key of A and e is the corresponding public key, and c is a random challenge:

$$
\begin{array}{lcl}
A & & B \\
 & \xleftarrow{\quad c \quad} & c \leftarrow c(R_B) \\
z \leftarrow c^w \mod n & \xrightarrow{\quad z \quad} & \\
 & & c \stackrel{?}{=} z^e \mod n
\end{array}
$$

Weakness: the signed texts are chosen solely by B, and this may allow the verifier to mount chosen-text attacks.

Motivation, II

If the same scheme (e.g. RSA) is used for both entity authentication and signatures, it is essential to use *different keys* for both purposes. Using the same key may compromise the security by allowing a chosen-text attack.

Suppose authentication consists of B challenging A with a random number r RSA-encrypted under A's public key, and A is required to respond with the decrypted random number. If B challenges A with $r = H(x)$, A's response to to this authentication request may (unwittingly) provide to B its RSA signature on the hash value of the (unknown to A) message x).

Solution: or use different keys or let the prover A to contribute to the message to be signed.

Σ-Protocols. General Setting

Σ*-protocol* is a three-move protocol between two parties, "prover" A and "verifier" B, where the prover acts first. The prover and verifier are modeled as probabilistic polynomial time interactive Turing machines. Furthermore, the verifier is expected to send only uniformly chosen bits.

The protocol is denoted by (A, B). $a(\cdot)$, $c(\cdot)$, $z(\cdot)$ and $\phi(\cdot)$ are polynomial time algorithms and depend typically on R and on the purpose of the protocol. We say that (A, B) is a Σ*-protocol for relation* R.

Σ-Protocols. Inputs

Both principals know v; only A knows w. R_A [resp R_B] is the random input of A [resp B]. The pair $(v, w) \in R$, where $R \subset \{0,1\}^* \times \{0,1\}^*$ is a publicly known, typically (but not necessary) polynomial time verifiable relation. Let

$$\mathrm{RW}(v) := \{w : (v, w) \in R\}$$

(the set of *witnesses* for v; $|w| \le p(|v|)$ for some polynomial p). Let $\mathrm{RX} := \{x : \mathrm{RW}(x) \neq \emptyset\}$.

Σ-Protocols. Description

$$
\begin{array}{lcl}
\qquad A & & \qquad\qquad B \\
a \leftarrow a(v, w, R_A) & \xrightarrow{\quad a \quad} & \\
 & \xleftarrow{\quad c \quad} & c \leftarrow c(R_B) \\
z \leftarrow z(v, w, R_A, c) & \xrightarrow{\quad z \quad} & \\
 & & \phi(v, a, c, z) \stackrel{?}{=} \mathrm{accept}
\end{array}
$$

a: *initial message*. $t_A = |a|$ is the *authentication length*
c: *challenge*, $c \leftarrow \{0, 1\}^{t_{R_B}}$.
z: *reply* (may reuse a).

Finally, B invokes a polynomial time computable predicate ϕ to check whether the *conversation* (x, a, c, z) is *accepting*.

Discrete Logarithm Problem

Let G_q be a group of prime order q. Let $g \in G_q$, $g \neq 1$. For each $h \in G_q$ there is a unique $w \in \mathbb{Z}_q$ such that $g^w = h$. w is called the *discrete logarithm* of h wrt g.

Let $\mathcal{G}$ be a family of groups of prime order such that the group operations can be performed efficiently, group elements can be efficiently sampled with uniform distribution and group membership as well as equality of group members can be efficiently tested. Let Gen be a probabilistic polynomial time generator that on input 1^k outputs a description of a group $G_q \in \mathcal{G}$ (including the prime group order q), and two random elements $g \neq 1$, h from G_q (alternatively, Gen can choose random elements $g \neq 1$, $w \in \mathbb{Z}_q$ and then set $h = g^w$). Elements from G_q are represented with k bits. Gen is *invulnerable* if it is infeasible, given just a string v generated according to Gen, to compute a witness w.

Example If G_q is a subgroup of order q in $\mathbb{Z}_p^*$, then the description of G_q consists of two primes p and q. Usually, $|p| > 600$ and $|q| > 160$.

Schnorr Identification Scheme

Let $\mathcal{G}$ be a family of groups. Let $(G_q, g, w) \leftarrow \text{Gen}(1^k)$ and let $h := g^w$. Let $v = (G_q, g, h)$ be the common input, w is the private input to A. The corresponding (unique) witness is $w \in \mathbb{Z}_q$ such that $g^w = h$. The relation R consists of all such pairs.

$$\begin{array}{ccc}
A & & B \\
r \leftarrow \mathbb{Z}_q;\, a := g^r & \xrightarrow{\qquad a \qquad} & \\
 & \xleftarrow{\qquad c \qquad} & c \leftarrow \{0,1\}^{80} \\
z \leftarrow cw + r & \xrightarrow{\qquad z \qquad} & g^z \stackrel{?}{=} ah^c
\end{array}$$

Check: $g^z = g^{cw+r} = g^r (g^w)^c = ah^c$.

Schnorr: Efficiency

Schnorr scheme was originally designed for smart-card applications, both communication and on-line computation are minimized.

Instead of a, $H(a)$ may sent in the first step. Verification consists of checking that $H(g^z h^{-c}) = H(a)$. There is no known attack (but the brute force) against the case when just $t = 80$ least significant bits were transferred.

Communication complexity: $\approx t + t + 2t = 4t = 320$ bits.

On-line signature generation: one $2t \times t$ bit multiplication (and one t-bit addition). Random number generation and exponentiation can be done off-line, during the processor's idle time.

If the scheme is used only for identification, where the prover has to reply to the challenge in a few seconds, the security parameter t could be lowered, say, to 48 bits.

Special Soundness

Let $v \in \{0,1\}^*$ be a string. A pair of accepting conversations (v, a, c, z) and (v, a, c', z') with $c \neq c'$ is called a *collision*. Σ-protocol (A, B) for relation R has the *collision-property* iff the following holds. Given a collision for $\underline{v \in RX}$, there exists an efficient algorithm (depending on (A, B) and R) that on input of a collision for v outputs w such that $(v, w) \in R$.

Σ-protocol (A, B) for R satisfies *special soundness*, iff it has the collision-property and there exist no collisions for $x \notin RX$. If membership of RX can be tested efficiently, the collision-property implies special soundness.

Lemma Let (A, B) satisfy SS. If $v \notin RX$ then an arbitrary prover A^*'s success probability of leading the honest verifier B to acceptance is at most $1/2^{t_B}$.

Zero-Knowledge

Suppose A and B engage in an execution of their protocol on common input v. B is interested to verify that A holds a witness w (a proof or a secret key). *Zero-knowledge* means roughly that no matter how B behaves as a verifier, it will not learn any information that it could not have computed itself, even before the start of the protocol.

(A, B) is *honest verifier zero-knowledge* if it is easy to simulate conversations with an honest verifier. If, additionally, the simulator works by taking any uniformly chosen challenge c as input and outputs an accepting conversation where c is the challenge (an accepting conversation (v, a, c, z)), then (A, B) is said to be *special honest verifier zero-knowledge*.

HVZK protocols are useful, since the general ZK protocols are far less efficient. Also, HVZK is sufficient in a wide range of applications. There exist transformation methods for turning certain classes of HVZK protocols into ZK ones.

Witness Hiding

Schnorr scheme: Security

Special Soundness. Given two accepting conversations (v, a, c, z) and (v, a, c', z'), with $c \neq c'$, w is computed as

$$w \leftarrow \frac{z - z'}{c - c'} = \frac{(cw + r) - (c'w + r)}{c - c'} = \frac{(c - c')w}{c - c'} \ .$$

Additionally, if G_q bases on $\mathbb{Z}_p^*$, then B can efficiently decide whether a given $v \in RX$. Thus, in that case Schnorr scheme satisfies special soundness.

Special HVZK Select $c, z \leftarrow \mathbb{Z}_q$, compute $a \leftarrow g^z \cdot h^{-c}$. Then (v, a, c, z) is an accepting conversation with the right distribution.

The protocol is not known to be witness hiding or zero knowledge, but is conjectured to be secure against an arbitrary polynomially bounded verifier, if computing discrete logarithms is intractable.

Guillou-Quisquater Scheme

Let n be an RSA-modulus, and let e be a prime smaller than n with e relatively prime to $\phi(n)$. Let $h, w \in \mathbb{Z}_n^*$, such that $w^e \equiv h \pmod{n}$. Let $v = (n, e, h)$, $R = (v, w)$. An invulnerable generator Gen(1^k) generates (1) a random RSA-modulus n of the desired length k bits ($|p| \approx |q|$); (2) random prime $e < n$, $\gcd(e, \phi(n)) = 1$; (3) random $w \in \mathbb{Z}_n^*$. Let $h := w^e \mod n$.

$$
\begin{array}{rcl}
A & & B \\
r \leftarrow \mathbb{Z}_n^*; a := r^e \mod n & \xrightarrow{\quad a \quad} & \\
 & \xleftarrow{\quad c \quad} & c \leftarrow \mathbb{Z}_e \\
z \leftarrow w^c r \mod n & \xrightarrow{\quad z \quad} & \\
 & & z^e \stackrel{?}{=} a h^c \mod n
\end{array}
$$

Note that A does not need to know the factorization of n.

GQ scheme: Security

Collision Intractability Given two accepting conversations (v, a, c, z) and (v, a, c', z'), with $c \neq c'$, w is computed as

$$w \leftarrow \left(\frac{z}{z'}\right)^{\frac{1}{c-c'}} = \left(\frac{w^c r}{w^{c'} r}\right)^{\frac{1}{c-c'}} = \left(w^{c-c'}\right)^{\frac{1}{c-c'}} \quad .$$

Here, $\frac{1}{c-c'}$ is computed in $\mathbb{Z}_e^*$, and B checks additionally the primality of e. Thus, GQ satisfies the collision-property and any efficient attacker that could compute a collision given just (n, e, h), could be used to invert the RSA. On the other hand, collisions can occur if $x \notin RX$ and thus GQ does not satisfy special soundness.

Special HVZK Select $c \leftarrow \mathbb{Z}_e$, $z \leftarrow \mathbb{Z}_n^*$ and compute $a \leftarrow z^e \cdot h^{-c}$. Then (v, a, c, z) is an accepting conversation with the right distribution.

GQ has not been proven to be secure against adaptive attacks (although it has been conjectured to be): it is not known to be witness hiding or ZK.

Okamoto-Schnorr Scheme

Let $\mathcal{G}$ be a family of groups. This time,

$$(G_q, g_1, g_2, w_1, w_2) \leftarrow \mathsf{Gen}(1^k) \ ,$$

where both g_1 and g_2 are generators. Let

$$h := g_1^{-w_1} g_2^{-w_2} \ .$$

Let $v = (G_q, g_1, g_2, h)$ be the common input, $w = (w_1, w_2)$ is the private input to A.

$$\begin{array}{lcr}
A & & B \\
r_1 \leftarrow \mathbb{Z}_q; r_2 \leftarrow \mathbb{Z}_q & & \\
a := g_1^{r_1} g_2^{r_2} & \xrightarrow{\quad a \quad} & \\
 & \xleftarrow{\quad c \quad} & c \leftarrow c(R_B) \\
z_1 \leftarrow cw_1 + r_1 & & \\
z_2 \leftarrow cw_2 + r_2 & \xrightarrow{\quad (z_1, z_2) \quad} & \\
 & & g_1^{z_1} g_2^{z_2} h^c \stackrel{?}{=} a
\end{array}$$

Check: $g_1^{z_1} g_2^{z_2} h^c = g_1^{cw_1+r_1} g_2^{cw_2+r_2} g_1^{-cw_1} g_2^{-cw_2} = g_1^{r_1} g_2^{r_2} = a$.

Okamoto-Schnorr Scheme: Security

Conversion to Signature Scheme

Step I

Let H be a random oracle: a function with initially empty database (a, c), such that $H(a)$ returns c if (a, c) is in the database for some c. Otherwise H generates uniformly a new c, adds (a, c) to the database and returns newly generated c.

Σ-protocols can be converted into signature schemes by using the next general method:

$$
\begin{array}{llcll}
A & a \leftarrow a(v, w, R_A) & \xrightarrow{\quad a \quad} & & B \\
 & & \xleftarrow{\quad c \quad} & c \leftarrow H(a) & \\
 & z \leftarrow z(v, w, R_A, c) & \xrightarrow{\quad z \quad} & & \\
 & & & \phi(v, a, c, z) \text{ accepts?} &
\end{array}
$$

Conversion to Signature Scheme

Step II

c is a random string that depends provably on the value a (exactly what was needed from the c!). Additionally, A can compute c herself and thus, interaction with B becomes unnecessary.

$$
\begin{array}{lcr}
A & & B \\
a \leftarrow a(v, w, R_A) & & \\
z \leftarrow z(v, w, R_A, H(a)) & \xrightarrow{(a, z)} & \\
 & & \phi(v, a, H(a), z) \text{ accepts?}
\end{array}
$$

Conversion to Signature Scheme

Step III

Finally, add a message m to be signed:

$$
\begin{array}{lcr}
A & & B \\
a \leftarrow a(v, w, R_A) & & \\
c \leftarrow H(m, a) & & \\
z \leftarrow z(v, w, R_A, c) & & \\
 & \xrightarrow{(m, a, c, z)} & \\
 & & c \stackrel{?}{=} H(m, a) \\
 & & \phi(v, a, c, z) \text{ accepts?}
\end{array}
$$

Schnorr Signature Scheme

Let $\mathcal{G}$ be a family of groups. Let $(G_q, w, h) \leftarrow \text{Gen}(1^k)$ and let $h := g^w$. Let $v = (G_q, g, h)$ be the common input, w is the private input to A. The corresponding (unique) witness is $w \in \mathbb{Z}_q$ such that $g^w = h$. The relation R consists of all such pairs.

$$
\begin{array}{lcr}
A & & B \\
r \leftarrow \mathbb{Z}_q; a := g^r & & \\
c \leftarrow H(m, a) & & \\
z \leftarrow cw + r & \xrightarrow{(m.a, c, z)} & \\
 & & c \stackrel{?}{=} H(m, a) \\
 & & g^z \stackrel{?}{=} ah^c
\end{array}
$$

Check: $g^z = g^{cw+r} = g^r(g^w)^c = g^w h^c = ah^c$.

Efficiency A has to perform on-line one H evaluation, one 160-bit multiplication and one addition. Communication can be reduced: A sends (m, c, z) and B verifies that $s = H(m, g^z h^{-c})$.

Caveats

H can be chosen to be a standard hash function, but in such case the conversion scheme loooses provable security (cf the original paper of Schnorr).

If both identification scheme and signature are used in the same smartcard, some care has to be taken. Namely, during the identification scheme B can output as the challenge $c = H(m, a)$ for m chosen by her. After receiving z from A, B will own a legitimate signature (a, c, z) of m.

Solution (Schnorr scheme): A sends the 80 least significant bits of a during the step 1. There is no known attack in this case.

Security Comparison of Schemes

Efficiency Comparison

r (random number generation), e (exponentiation), m (multiplication), a (addition), h (hashing).

	Schnorr	GQ	Okamoto-Schnorr
Signer (off-line)	$r + e$	$r + e$	$2r + 2e + m$
Verifier (off-line)	r	r	r
Signer (on-line)	$m + a$	$e + m$	$2m + 2a$
Verifier	$2e + m$	$2e + m$	$3e + 2m$
Operations:	mod q	mod n	mod q

Usually, $n \approx 1024$ and $q \approx 160$. (Online) Schnorr and Okamoto-Schnorr schemes are faster than GQ by several orders of magnitude.

RSA identification is very close to the GQ scheme in performance.

Patent issues

GQ Scheme patented by U.S. Phillips Corporation. US Patent 5140634: Method and apparatus for authenticating accreditations and for authenticating and signing messages”

http://patent.womplex.ibm.com/details?patent_number=5140634

Schnorr Scheme patented by RSADSI, US Patent 4995082 ”Method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system.”

http://patent.womplex.ibm.com/details?patent_number=4995082

Restrictive (RSADSI has claimed that DSA violates the patent).

Smart-cards

More Applications

Aside from identification and signing, Σ-protocols are also extensively used in the following areas:

- Blind signature/digital cash protocols. For example, Pointcheval-Stern provably secure blind signatures is based on Okamoto-Schnorr identification scheme.

- Electronic voting. For example, Cramer-Gennaro-Schoenmakers secure and optimally efficient election scheme is based on Schnorr identification scheme.