T-79.511 Special Course on Cryptology / Zero Knowledge: Zero Knowledge Proofs of Identity and Proofs of Knowledge

Jan Hlinovsky
jan.hlinovsky@hut.fi

October 31, 2001

1 Introduction

Authentication or proving one's identity can be done in many ways, but a typical way in applications related to computers has been the use of passwords. A big disadvantage in using passwords is, that the party who is verifying the authentication (called the verifier) or anyone eavesdropping the communication can later impersonate the original authenticator (called the prover).

A more advanced way for authentication is challenge-response method, where a prover demonstrates the knowledge of a secret by responding to the verifier's challenge in a way that is not directly reusable by the verifier (e.g. encrypt a random challenge with a secret key). This method, however, might reveal something about the secret, especially so if the verifier can choose the challenges that he sends (chosen text attack) [9]. So, the idea of zero knowledge protocols seems to be quite useful and natural in this context.

In this survey, we will briefly look at zero knowledge proofs of knowledge, and more carefully special kind of zero knowledge proofs of knowledge, namely zero knowledge proofs of identity. We will examine more closely a specific protocol by Feige, Fiat and Shamir and prove that it is indeed a zero knowledge proof of identity.

2 Proofs of knowledge

The concept "proof of knowledge" appeared first in the seminal paper of Goldwasser, Micali and Rackoff [7], but they did not give a formal definition for it. Formal definitions appear in [5] and [12], but Bellare and Goldreich point out some problems with these definitions and suggest a better one in [1]. The differences can be quite subtle, as the following quote perhaps suggests: "Intuitively, a two-party protocol constitutes a "system for proofs of knowledge" if "whenever" one party (called the verifier) is "convinced" then the other party (called the prover) indeed "knows" "something". The excessive use of quotation symbols in the condition of the above statement may provide some indication to the complexity of the notion." [1]

Let's first have a look at the definition of Feige, Fiat and Shamir[5]. The prover P and verifier V are polynomial time probabilistic Turing machines that have a common input tape I, two common tapes CP and CV, private work tapes WP and WV, and private random tapes RP and RV. The prover has an additional private tape S ("the secret").

Definition 2.1 [5] A pair of interacting polynomial time probabilistic Turing machines P and V is called an *interactive proof system of knowledge* for the polynomial time predicate $p(I, S)$ if:

1. *(Completeness)* For all I for which $p(I, S)$ is satisfiable, the execution of (P, V) on input I succeeds with overwhelming probability.
2. *(Soundness)* There exists an M (with complete control over cheater P') such that for all (polynomial time cheaters) P' , RP and sufficiently large I, if the execution of (P', V) on input I succeeds with non negligible probability, then the output produced by M at the end of the execution of (P', M) on input I satisfies the predicate with overwhelming probability.

Definition 2.2 [5] Let V' be an arbitrary polynomial time program that tries to extract additional information from P. An interactive proof system of knowledge is *zero knowledge* if for all V' its view of the communication in (P, V') can be recreated with an indistinguishable probability distribution under no additional assumptions by a polynomial time probabilistic Turing machine M.

Note that both the prover and the verifier are polynomial time probabilistic Turing machines, so these proof systems are *arguments*. This limitation seems practical, because a unlimited prover has naturally access to e.g. information whether $x \in L$, where L is some language in, say, **NP**. Bellare and Goldberg, however, argue that "for the purpose of *defining* proofs of knowledge, there is no need to restrict the prover to polynomial-time" [1], and their definition indeed does not assume anything from the power of the prover.

The definition by Bellare and Goldreich is in many ways more generally applicable, they don't even require soundness. For the purposes of the identification schemes in this survey, however, the definition of [5] is sufficient, and the reader is advised to see [1] if interested in a more general definition.

3 Feige-Fiat-Shamir identification scheme

Informally defined, an *identification scheme* (or *proof of identity*) is a protocol which enables P to prove her identity polynomially many times to V without enabling V to impersonate P.

We now present an identification scheme by Feige, Fiat and Shamir. The authors of the scheme claim that with reasonable paramater choises the protocol is about two orders of magnitude faster than RSA-based identity schemes. The security of the scheme relies on the assumption that factoring is difficult (actually, the problem is taking square root mod n, where n is a large integer whose factorization is unknown, but this is computationally equivalent to factoring n [9]).

We need a trusted center T that publishes a modulus $n = pq$, where p and q are large secret primes congruent to to 3 mod 4. Thus n is a so called Blum integer, and -1 is a quadratic non-residue (mod n) whose Jacobi symbol is $+1$. The secrecy of p and q is crucial to the security of the system and the TTP-service is needed in this scheme. The center also publishes the parameteres k (the key size) and t (number of protocol iterations).

Next the prover P needs to generate a key pair as follows:

1. Select k random numbers $S_1, \ldots, S_k$ in $\mathbb{Z}_n$
2. Choose each I_j as $(-1)^b S_j^{-2}$ $(mod\ n)$, where b is a random bit (0 or 1).
3. Publish $I_1, \ldots, I_k$ (the public key) and keep $S_1, \ldots, S_k$ (the private key) secret.

Now the core protocol is executed t times:

1. P picks a random integer r and a random bit b, and sends $X = (-1)^b r^2 (mod\ n)$ to V

   (In practice V should check here that $X \neq 0$. If $X = 0$, V should quit or ask for a new X.)
2. V sends a vector of random bits $(E_1, \ldots, E_k)$ to P
3. P computes $Y = r \cdot \prod_{E_j=1} S_j (mod\ n)$ and sends it to V
4. V verifies that $X = \pm Y^2 \cdot \prod_{E_j=1} I_j (mod\ n)$

The error probability of the identification scheme is 2^{-kt}.

Theorem 3.1 *[5]: This protocol is zero knowledge proof of knowledge (as defined in definitions 2.1 and 2.2) of the S_j for $k = O(log\ log\ n)$ and $t = O(log\ n)$*

Proof sketch.
Completeness:

$$Y^2 \cdot \prod_{E_j=1} I_j (mod\ n) = \left(r \cdot \prod_{E_j=1} S_j \right)^2 \cdot \prod_{E_j=1} I_j = r^2 \cdot \prod_{E_j=1} (S_j^2 I_j) = \pm r^2 = \pm X (mod\ n) \quad (1)$$

This shows that the real prover can always convince the verifier.

Soundness:

Next we want to show that whenever V accepts a cheater P':s proof with non negligible probability, M can print out all the S_j with overwhelming probability.

We first create an execution tree of (P', V) for input I and random tape RP. At each of the t stages of the protocol, V can ask $2^k = (log\ n)^{O(1)}$ possible questions, so that each node in the tree may have only polynomially many children in terms of $|I|$. Next we truncate the tree. Any child that corresponds to a problem that P' does not answer correctly is eliminated, as well as its descendants. What remains is then a tree where the paths from root to leaves correspond to the succesfull executions of the protocol. Let us call this tree T.

We call a node heavy if it's degree is larger than $2^k/2$, i.e. in T a node is heavy if more than half of the executions in this state are successful. We now want to show that all the S_j can be computed from the sons of a heavy node and that a polynomial time M can find a heavy node in T with overwhelming probability.

With a heavy node H let Q be the set of bit-vector queries $(E_1, \ldots, E_k)$ which are properly answered by P'. It can be shown that Q, which has more than $2^k/2$ bit-vectors of length k must contain two vectors $(E'_1, \ldots, E'_k)$ and $(E''_1, \ldots, E''_k)$ in which $E'_j = 0, E''_j = 1, 1 \leq j \leq k$ and $E'_i = E''_i, i \neq j$.

Because both of these queries are properly answered, the verification part of the protocol tells us that

$$X' = \pm Y'^2 \cdot \prod_{E'_j=1} I_j (mod\ n)$$

$$X'' = \pm Y''^2 \cdot \prod_{E''_j=1} I_j (mod\ n)$$

Now because P' had to choose X before obtaining the query from V, it must hold that $X' = X''$.

The equations above can be manipulated to get $(Y''/Y')^2 = \pm 1/I_j (mod\ n)$. Because n is a Blum integer only one of $1/I_j$ and $-1/I_j$ has a square root, we conclude that Y''/Y' is the desired S_j.

Next it is shown that there must be at least one level in T where at least half of the nodes are heavy. Let α_i be the ratio between the number of nodes at level $i+1$ and level i in T. If $\alpha_i \leq (3/4)2^k$ for all $1 \leq i \leq t$, then the total number of leaves $\prod_i \alpha_i$ is bounded by $(3/4)^t 2^{kt}$, which is a negligible fraction of the 2^{kt} possible leaves. But because we assume that this fraction is polynomial, it must be that $\alpha_i > (3/4)2^k$ for at least one i, and thus at least half of the nodes at this level must be heavy.

To find a heavy node, M will do repeated resets and executions of P' to determine the degrees of polynomially many random nodes at each level.This can be done in polynomial time, because the fraction of the possible leaves that appear in the truncated tree is assumed to be non negligible.

Zero knowledge: We see that a false prover P' can cheat V with probability 2^{-k} by guessing the challenge vector $(E_1, \ldots, E_k)$. Using resettable simulation, M can mimic the communication in (P, V') with an indistinguishable probability distribution in $O(t \cdot 2^k)$ expected time, which is polynomial by our assumptions on k and t. This would be enough to satisfy the definition of zero knowledge in [7], but to satisfy the definition 2.2 in this survey we need to add that the existence of S_j which satisfy $I_j S_j^2 = \pm 1 (mod\ n)$ is guaranteed for any I_j with Jacobi symbol $+1$ and this property is checkable in polynomial time. ∎

4 Parallelisation of the protocol

A natural way get the error probability small enough is to repeat the basic protocol either secuentially or in parallel. Parallel execution would be very practical, since the number of rounds is constant. The basic protocol of Feige, Fiat and Shamir has kind of a built-in parallelism (vereifier asks k questions simultaneously) but this is limited as shown above. In general, the parallel execution of the protocol is not zero knowledge. The authors claim, however, that the information that is revealed in parallel execution is "unusable" and they define it in a following way:

Definition 4.1 [5] The protocol (P, V) releases no transferable information if it succeeds with overwhelming probability but there is no coalition of P',V' with the property that after polynomially many executions of (P, V') it is possible to execute (P', V) with a non negligible probability of success.

The authors of [5] then prove that if factoring is difficult, the parallel version of their identification scheme releases no transferable information.

5 Other identification schemes

The Feige-Fiat-Shamir identification scheme can be seen as a refined version of an earlier scheme by Fiat and Shamir[6]. Many others have also made extensions and refinements to that protocol, one example is the Guillou-Quisquater (GQ) identification scheme [8]. Chaum, Evertse and Graaf have made an interactive protocol for proving posession of a discrete logarithm [2], and their ideas have been extended e.g. by Schnorr [10] for both identification and signature generation. The Schnorr scheme is not zero knowledge but it is a proof of knowledge. The Schnorr scheme seems to be quite famous, perhaps because of a preprocessing scheme for reducing real time computation that was broken by de Rooj [3]. The scheme was fixed by Schnorr [11] and broken again by de Rooj [4].

6 Conclusions

There are several different definitions of proofs of knowledge, here we use the one that is defined in [5]. Bellare and Goldreich have pointed out several limitations in this definition and given a more general definition [1] but their criticism does not change the validity of the results presented in [5] and here.

Zero knowledge proofs of identity are a special case from proofs of knowledge. They can be used for (entity) authentication in a quite natural way, with great advantages compared to e.g. using passwords. An example identification scheme by Feige, Fiat and Shamir is presented, and this is proved to be a proof of knowledge and also zero knowledge provided certain parameter limitations hold. A trusted third party is needed for connecting the "real identity" with the public key.

The proofs of identity discussed here are actually arguments, since they limit the prover to be polynomial time, and in addition to that many of them rely on some unproven assumptions, e.g. that factoring or calculating discrete logarithm is hard.

References

[1] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. Advances in Cryptology–CRYPTO 92. Springer-Verlag, 1992.

[2] D. Chaum, J.-H. Evertse, and J. van de Graaf, An improved protocol for demonstrating posession of discrete logarithms and some generalizations. Advances in Cryptology – EUROCRYPT '87 (D. Chaum and W.L. Price, eds.), Lecture Notes in Computer Science, vol. 304, Springer-Verlag, 1987, pp. 127–142.

[3] P. de Rooij, On the security of the Schnorr scheme using preprocessing. Proc. of Eurocrypt'91, LNCS series, Springer-Verlag, 1991.

[4] P. de Rooij, On Schnorr's preprocessing for digital signature schemes. Journal of Cryptology, 10 (1997), 1-16.

[5] Uriel Feige, Amos Fiat, and Adi Shamir, Zero-knowledge Proofs of Identity. Proceedings of the 19th Annual ACM Symposium on Theory of Computing, New York, 1987.

[6] Amos Fiat and Adi Shamir, How to prove yourself: Practical solutions to identification and signature problems. CRYPTO 86, 1987.

[7] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In Proceedings of the 17th ACM Symposium on the Theory of Computing STOC, 1985. ACM Press.

[8] L. Guillou and J.-J. Quisquater, A Practical Zero-Knowledge Protocol fitted to Security Microprocessor Minimizing both Transmission and Memory. Proc. of EuroCrypt 88, Springer Verlag LNCS series.

[9] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.

[10] C. P. Schnorr, Efficient identification and signatures for smart cards. Proc. of Crypto'89, Lect. Notes in Comp. Sci., Springer-Verlag, Berlin, 435 (1990), 239-252.

[11] C. P. Schnorr, Efficient signature generation by smart cards. Journal of Cryptology, 4 (1991), 161-174.

[12] M. Tompa and H. Woll, Random Self Reducibility and Zero-knowledge Interactive Proofs of Knowledge, Proceedings of the 28th Symposium on the Foundation of Computer Science, 1987.