CMSC 858K — Advanced Topics in Cryptography	April 29, 2004
Lecture 25	
Lecturer: Jonathan Katz	*Scribe(s): (none)*

1 Public-Key Identification Schemes

In this lecture, we discuss concrete applications of zero-knowledge proof systems to the task of constructing (public-key[1]) identification schemes. We begin with an informal definition of what these are.

Definition 1 A (public-key) identification scheme consists of a prover P who has generated some public-/private-key pair (PK, SK), and wants to prove his identity to a verifier V who knows PK. ♢

In terms of security, we say a scheme is secure against a *passive eavesdropper* if an adversary cannot impersonate the prover even after passively monitoring multiple executions of the protocol between the (real) prover and the (honest) verifier. We say a scheme is secure against an *active adversary* if an adversary cannot impersonate the prover (to an honest verifier) even after the adversary — acting in the role of the verifier — has interacted with the prover multiple times. We stress that in the latter case the adversary is not restricted to running the honest verification protocol, but may instead deviate arbitrarily from the prescribed actions. We also remark that security against an active adversary implies security against passive eavesdropping, since an active adversary can simulate passive eavesdropping by simply acting as an honest verifier.

It is simple to construct a public-key identification scheme from any signature scheme:

$$\begin{array}{lcl}
\underline{P(SK)} & & \underline{V(PK)} \\
 & \xleftarrow{\quad r \quad} & r \leftarrow \{0,1\}^k \\
\sigma \leftarrow \mathrm{sign}_{SK}(r) & \xrightarrow{\quad \sigma \quad} & \text{accept if } \mathrm{Verify}_{PK}(r,\sigma)=1
\end{array}$$

This scheme is secure against an active adversary. As an informal proof, imagine an adversary who — playing the role of the verifier — interacts with the honest prover, say, n times. In doing so, the adversary may send whatever values $r_1, \ldots, r_n$ it likes to the prover, who responds by signing these values. Then, the adversary interacts with an honest verifier who sends some value $\tilde{r}$. There are two possibilities:

1. $\tilde{r} \in \{r_1, \ldots r_n\}$. This happens only with probability $n/2^k$. Since n is polynomial in the security parameter k, this probability is negligible.

2. $\tilde{r} \notin \{r_1, \ldots r_n\}$. In this case, impersonation of the prover is equivalent to forging a signature on a "new" (i.e., previously-unsigned) message. By security of the signature scheme, this is infeasible.

[1] Although it is possible to consider the analogous case of private-key identification, the public-key setting is the one more often considered in this context. Thus, when we say "identification scheme" we mean the public-key setting by default.

It is also possible to construct an identification scheme based on any CCA1-secure public-key encryption scheme:

$$
\begin{array}{lcl}
\underline{P(SK)} & & \underline{V(PK)} \\
 & \xleftarrow{\quad \varepsilon_{PK}(r) \quad} & r \leftarrow \{0,1\}^k \\
 & \xrightarrow{\quad r' \quad} & \text{Accept if } r = r'
\end{array}
$$

A proof of security for this protocol (against active attacks) is slightly more difficult, but not much more so.

Although these schemes are conceptually simple and achieve security against active attacks, there are at least two drawbacks that have motivated the search for other identification schemes: (1) the above schemes leave an undeniable trace that the prover executed the identification protocol. One may want to avoid leaving behind such "evidence". (2) Generally speaking, signature schemes and CCA1-secure encryption schemes are difficult to construct (especially if we limit ourselves to the standard model). Thus, it is natural to wonder whether we can construct identification schemes without relying on these tools. Indeed, as we will briefly mention below, we can go in the reverse direction and build efficient signature schemes (in the random oracle model) from identification schemes of a certain form. In fact, it is fair to say that identification protocols have almost no practical application as identification protocols *per se*. (The reason is that one typically wants identification to be coupled with key exchange, so that communication between the parties can be secured once they have identified each other.) However, identification protocols have found extensive application as building blocks for other primitives, with signature schemes being the primary example.

2 General Paradigms for Constructing Identification Schemes

We show now some general techniques for constructing identification schemes based on proof systems satisfying various additional properties (many of these ideas are due to Feige, Fiat, and Shamir [1]). One immediate idea is to use a *zero-knowledge (ZK) proof of knowledge (PoK)* for a "hard" language. In more detail, consider the following identification protocol based on a one-way function f:

- The public and secret keys are generated as follows: the prover chooses a random $x \in \{0,1\}^k$ (where k is the security parameter) and sets $y = f(x)$. The public key is y and the secret key is x.
- To identify himself to a verifier holding his public key y, the prover gives a ZK PoK (with negligible soundness error) of x s.t. $f(x) = y$. The verifier accepts iff the proof is successful.

We remark that, if we don't care about round complexity, the ZK PoK can be based on the one-way function f and so we get an identification protocol based on any one-way function.

It is not too hard to see that the above gives an identification scheme secure against active attacks. We do not give a formal proof, but instead describe informally how such a proof would proceed.

Theorem 1 *The above identification scheme is secure against active adversaries.*

Proof (Sketch) Assume we have an adversary A attacking the above identification scheme via an active attack, and succeeding with probability $\varepsilon(k)$. We will prove that $\varepsilon(k)$ is negligible. We can view the adversary's attack as follows: first, the adversary receives a public key y, where $y = f(x)$ for random $x \in \{0,1\}^k$. Next, the adversary — playing the role of a verifier who may act in an arbitrary manner — interacts with the honest prover as many times as the adversary likes. We call this stage 1. Finally, the adversary interacts with an honest verifier (holding public key y); we call this stage 2. The adversary succeeds in impersonating the prover (in stage 2) with probability $\varepsilon(k)$, where this probability is taken over y, random coins used by the adversary and the honest prover in stage 1, and the random coins of the adversary and the honest verifier in stage 2.

Now, consider modifying the above experiment in the following way: in stage 1, instead of having the honest prover P perform a real execution of the protocol with the adversary A in stage 1, we instead run the zero-knowledge simulator Sim (guaranteed for the proof system) in stage 1. Let $\varepsilon'(k)$ denote the probability that A succeeds in impersonating P in stage 2 in this modified experiment. By the properties of the ZK simulator, we can show that the difference $|\varepsilon(k) - \varepsilon'(k)|$ is negligible.

In the previous experiment, we no longer need to know a pre-image of y (i.e., x from the original experiment). This means we can construct an adversary A' (who will internally run both A and Sim) who obtains a value y and then succeeds in giving a valid ZK PoK for this y with probability $\varepsilon'(k)$ (in particular, phase 1 is no longer relevant since A' is simulating all of phase 1 internally and there is no longer any need to interact with an outside prover).

But now we claim that we can use such an A' to invert the one-way function f on a randomly-generated output point y (i.e., $y = f(x)$ for random x) with probability negligibly close to $\varepsilon'(k)$ in case this latter quantity is non-negligible. How do we do this? We simply run the *knowledge extractor* for this proof system with adversarial prover A'. This extractor guarantees (informally) that if $\varepsilon'(k)$ is non-negligible — and, in particular, larger than the soundness error of the proof system — then an inverse of y will be extracted with probability negligibly close to $\varepsilon'(k)$.

In summary, if $\varepsilon(k)$ is non-negligible, we end up with an efficient algorithm (but see the remark below) inverting f with non-negligible probability. Since f is a one-way function, this is a contradiction. Thus, we must have $\varepsilon(k)$ negligible, as desired.

Technical remark: Actually, the algorithm we construct to invert f runs in *expected polynomial time* rather than *strict polynomial time* (this is so since both the ZK simulator and the PoK knowledge extractor may run in expected polymomial time). However, one can show that the existence of an expected polynomial-time algorithm A_1 which inverts f with non-negligible probability implies the existence of a strict polynomial-time algorithm A_2 which inverts f with non-negligible probability (without giving the details, A_2 simply runs A_1 but aborts if it runs for "too long"). Thus, the above proof does indeed lead to a contradiction. ∎

The above construction can be simplified if we are content to achieve security against a passive adversary. In this case, we only need a proof system which is *honest verifier* zero-knowledge (as well as being a proof of knowledge). Since in the case of a passive attack the adversary (by definition) is limited to eavesdropping on executions of the protocol between the prover and an *honest* verifier, honest-verifier zero-knowledge suffices in this case.

2.1 On Using Witness Indistinguishability

A natural question is whether we can further simplify things by using a *witness indistinguishable* (WI) proof of knowledge (in the case of an active adversary). As we will see, the answer is *yes*; however, it requires some changes to the protocol as stated above. In particular, recall that a WI proof guarantees nothing in case there is only one witness (since then witness indistinguishability is trivial to achieve); so, we need to modify the protocol so that two or more witnesses are available. The idea is similar to that used by Feige-Shamir (cf. the previous lecture) in constructing their constant-round ZK argument of knowledge.

Let f be a one-way function. The protocol is as follows:

- The public and secret keys are generated by having the prover chooses *two* random points $x_1, x_2 \in \{0,1\}^k$ and setting $y_1 = f(x_1)$ and $y_2 = f(x_2)$. The public key is (y_1, y_2) and the secret key is x_1. (As we will see, x_2 is not needed.)
- To identify himself to a verifier holding his public key (y_1, y_2), the prover gives a WI PoK (with negligible soundness error) of x such that either $f(x) = y_1$ or $f(x) = y_2$.

As before, we provide only a sketch of security for the above protocol.

Theorem 2 *The above identification scheme is secure against active adversaries.*

Proof (Sketch) Assume we have an adversary A attacking the above identification scheme via an active attack, and succeeding with probability $\varepsilon(k)$. We will prove that $\varepsilon(k)$ is negligible. We can view the adversary's attack as follows: first, the adversary receives a public key (y_1, y_2), where $y_i = f(x_i)$ for random $x_i \in \{0,1\}^k$. Next, the adversary — playing the role of a verifier who may act in an arbitrary manner — interacts with the honest prover as many times as the adversary likes. We call this stage 1. Finally, the adversary interacts with an honest verifier (holding public key (y_1, y_2)); we call this stage 2. The adversary succeeds in impersonating the prover (in stage 2) with probability $\varepsilon(k)$, where this probability is taken over (y_1, y_2), random coins used by the adversary and the honest prover in stage 1, and the random coins of the adversary and the honest verifier in stage 2.

If $\varepsilon(k)$ is not negligible, then by negligible soundness error of the PoK we can extract a witness from the adversary in stage 2 of its attack with probability negligibly close to $\varepsilon(k)$. In particular, we extract either a "type 1 witness" (i.e., an x such that $f(x) = y_1$) or a "type 2 witness" (i.e., an x such that $f(x) = y_2$) with non-negligible probability. Let $\varepsilon_1(k)$ denote the probability that we extract a type 1 witness, and similarly for $\varepsilon_2(k)$. The above shows that either $\varepsilon_1(k)$ or $\varepsilon_2(k)$ is not negligible (or possibly both are). We show that either of these lead to an efficient procedure for inverting f with non-negligible probability, and hence a contradiction.

If $\varepsilon_2(k)$ is not negligible, we can almost immediately derive our desired contradiction. Given a point y which is equal to $f(x)$ for a random x, simply choose $x_1 \in \{0,1\}^k$ at random and set the public key equal to (y_1, y). We can now interact with the adversary exactly as in the real experiment (recall that in the real protocol, the prover does not use

x_2 anyway) and extract a type 2 witness with probability exactly $\varepsilon_2(k)$. (Note that this will exactly be an inverse of y.) Since $\varepsilon_2(k)$ is non-negligible by assumption, we have the desired contradiction.

In case $\varepsilon_1(k)$ is non-negligible, the only thing we need to do is to "switch" from using x_1 to using x_2. In particular, we now construct the public key (y_1, y_2) in the same way as before, but throw away x_1 instead of x_2. Also, when interacting (as a prover) with the adversary A (acting as a verifier) we use witness x_2 instead of witness x_1. (We extract a witness from the adversary in stage 2 as before.) By witness indistinguishability of the proof system, one can show that the probability $\varepsilon'_2(k)$ of extracting a type 2 witness is negligibly-close to the probability of extracting a type 2 witness in the original experiment (i.e., $|\varepsilon_1(k) - \varepsilon_2(k)|$ is negligible). Thus, $\varepsilon'_2(k)$ is non-negligible. That this leads to a contradiction can be proven in exactly the same way as above.

Technical remark: The same issue arises here as in the previous proof, in that the adversary we construct to invert f runs in expected polynomial time rather than strict polynomial time (due to the possible expected-polynomial running time of the knowledge extractor). We may deal with this in exactly the same way as before. ■

In a later lecture, we will show how to *efficiently* instantiate the protocols of this section and the previous section based on a specific number-theoretic assumption.

2.2 On Avoiding Proofs of Knowledge

The constructions in the previous sections rely in an essential way on the use of proofs of knowledge (examining the previous proofs, we see that we need this property in order to be able to extract the desired inverse of f). We show here that it is possible to base identification schemes on zero-knowledge *proofs*.[2] We will only sketch the idea, and leave the proof to the reader. As far as we are aware, this idea originates in [2].

The concept is quite simple: instead of proving knowledge of some hard-to-compute function (namely, $f^{-1}(y)$ as in the previous sections), we prove membership in some hard-to-decide language. To be specific, let G be a pseudorandom generator (PRG) stretching its input by k bits (we know that this can be constructed based on any one-way function). Then consider the following identification protocol:

- The public and secret keys are generated by choosing a random $x \in \{0,1\}^k$ and then setting $y = G(x)$. The public key is y and the secret key is x.
- To identify himself to a verifier holder his public key y, the prover gives a ZK proof that $y = G(x)$ for some x (i.e., that y is in the image of G, or that y is pseudorandom).

One can show that the above identification scheme is secure against active adversaries.

[2] We believe it is possible to also base identification schemes on witness indistinguishable proofs; however, the idea of the previous section applied to the protocol of the present section will not work, and a more complex construction is required.

References

[1] U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. *J. Crypto* 1(2): 77–94 (1988).

[2] J. Katz and N. Wang. Efficiency Improvements for Signature Schemes with Tight Security Reductions. ACM CCCS 2003.