Identification Protocols Secure Against Reset Attacks

Mihir Bellare[1], Marc Fischlin[2], Shafi Goldwasser[3], and Silvio Micali[3]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA.
E-mail: mihir@cs.ucsd.edu.
URL: www-cse.ucsd.edu/users/mihir.
[2] Dept. of Mathematics (AG 7.2), Johann Wolfgang Goethe-University, Postfach 111932, 60054 Frankfurt/Main, Germany.
E-mail: marc@mi.informatik.uni-frankfurt.de
URL: www.mi.informatik.uni-frankfurt.de
[3] MIT Laboratory for Computer Science, 545 Technology Square, Cambridge MA 02139, USA.

Abstract. We provide identi£cation protocols that are secure even when the adversary can reset the internal state and/or randomization source of the user identifying itself, and when executed in an asynchronous environment like the Internet that gives the adversary concurrent access to instances of the user. These protocols are suitable for use by devices (like smartcards) which when under adversary control may not be able to reliably maintain their internal state between invocations.

1 Introduction

An identi£cation protocol enables one entity to identify itself to another as the legitimate owner of some key. This problem has been considered in a variety of settings. Here we are interested in an asymmetric setting. The entity identifying itself is typically called the prover, while the entity to which the prover is identifying itself is called the veri£er. The prover holds a secret key sk whose corresponding public key pk is assumed to be held by the veri£er.

The adversary's goal is to impersonate the prover, meaning to get the verifier to accept it as the owner of the public key pk. Towards this goal, it is allowed various types of attacks on the prover. In the model of smartcard based identification considered by [11], the adversary may play the role of verifier and interact with the prover, trying to learn something about sk, before making its impersonation attempt. In the model of "Internet" based identification considered by [6, 1, 5], the adversary is allowed to interact concurrently with many different prover "instances" as well as with the verifier. Formal notions of security corresponding to these settings have been provided in the works in question, and there are many protocol solutions for them in the literature.

In this work we consider a novel attack capability for the adversary. We allow it, while interacting with the prover, to reset the prover's internal state. That is, it can "backup" the prover, maintaining the prover's coins, and continue its interaction with the prover. In order to allow the adversary to get the maximum possible bene£t from this new capability, we also allow it to have concurrent access to different prover instances. Thus, it can interact with different prover instances and reset each of them at

will towards its goal of impersonating the prover. The question of the security of identi-£cation protocols under reset attacks was raised by Canetti, Goldreich, Goldwasser and Micali [8], who considered the same issue in the context of zero-knowledge proofs.

1.1 The power of reset attacks

AN EXAMPLE. Let us illustrate the power of reset attacks with an example. A popular paradigm for smartcard based identification is to use a proof of knowledge [11]. The prover's public key is an instance of a hard NP language L, and the secret key is a witness to the membership of the public key in L. The protocol enables the prover to prove that it "knows" sk. A protocol that is a proof of knowledge for a hard problem, and also has an appropriate zero-knowledge type property such as being witness hiding [12], is a secure identification protocol in the smartcard model [11].

A simple instance is the zero-knowledge proof of quadratic residuosity of [15]. The prover's public key consists of a composite integer N and a quadratic residue $u \in Z_N^*$. The corresponding secret key is a square root $s \in Z_N^*$ of u. The prover proves that it "knows" a square root of u, as follows. It begins the protocol by picking a random $r \in Z_N^*$ and sending $y = r^2 \bmod N$ to the verifier. The latter responds with a random challenge bit c. The prover replies with $a = rs^c \bmod N$, meaning it returns r if $c = 0$ and $rs \bmod N$ if $c = 1$. The verifier checks that $a^2 \equiv yu^c \bmod N$. (This atomic protocol has an error probability of $1/2$, which can be lowered by sequential repetition. The Fiat-Shamir protocol [13] can be viewed as a parallelized variant of this protocol.)

Now suppose the adversary is able to mount reset attacks on the prover. It can run the prover to get y, feed it challenge 0, and get back $a = r$. Now, it backs the prover up to the step just after it returned y, and feeds it challenge 1 to get answer $a' = rs$. From a and a' it is easily able to extract the prover's secret key s. Thus, this protocol is not secure under reset attacks.

Generalizing from the example, we see that in fact, all proof of knowledge based identi£cation protocols can broken in the same way. Indeed, in a proof of knowledge, the prover is de£ned to "know a secret" exactly when this secret can be extracted by a polynomial time algorithm (the "extractor") which has oracle access to the prover and is allowed to reset the latter [11, 4]. An attacker allowed a reset attack can simply run the extractor, with the same result, namely it gets the secret. So the bulk of ef£-cient smartcard based identi£cation protocols in the literature are insecure under reset attacks.

MOUNTING RESET ATTACKS. Resetting or restoring the computational state of a device is particularly simple in the case the device consists of a smartcard which the enemy can capture and experiment with. If the card is manufactured with secure hardware, the enemy may not be able to read its secret content, but it could disconnect its battery so as to restore the card's secret internal content to some initial state, and then re-insert the battery and use it with that state a number of times. If the smart card implements a proof of knowledge prover for ID purposes, then such an active enemy may impersonate the prover later on.

Other scenarios in which such an attack can be realized is if an enemy is able to force a crash on the device executing the prover algorithm, in order to force it to re-

sume computation after the crash in an older "computational state", thereby forcing it to essentially reset itself.

CAN WE USE RESETTABLE ZERO-KNOWLEDGE? Zero-knowledge proofs of membership secure under reset attack do exist [8], but for reasons similar to those illustrated above, are not proofs of knowledge. Accordingly, they cannot be used for identi£cation under a proof of knowledge paradigm. One of the solution paradigms we illustrate later however will show how proofs of membership, rather than proofs of knowledge, can be used for identi£cation.

1.2 Notions of security

Towards the goal of proving identification protocols secure against reset attacks, we £rst discuss the notions of security we de£ne and use.

We distinguish between two types of resettable attacks CR1 (Concurrent-Reset-1) and CR2 (Concurrent-Reset-2). In a CR1 attack, Vicky (the adversary) may interact concurrently, in the role of verifier, with many instances of the prover Alice, resetting Alice to initial conditions and interleaving executions, hoping to learn enough to be able to impersonate Alice in a future time. Later, Vicky will try to impersonate Alice, trying to identify herself as Alice to Bob (the verifier).

In a CR2 attack, Vicky, *while* trying to impersonate Alice (i.e attempting to identify herself as Alice to Bob the veri£er), may interact concurrently, in the role of veri£er, with many instances of the prover Alice, resetting Alice to initial conditions and interleaving executions. Clearly, a CR1 attack is a special case of a CR2 attack.

A de£nition of what it means for Vicky to win in the CR1 setting is straightforward: Vicky wins if she can make the veri£er Bob accept. In the CR2 setting Vicky can make the veri£er accept by simply being the woman-in-the-middle, passing messages back and forth between Bob and Alice. The de£nitional issues are now much more complex because the woman-in-the-middle "attack" is not really an attack and the de£nition must take this into account. We address these issues based on de£nitional ideas from [6, 5], speci£cally by assigning session-ids to each completed execution of an ID protocol, which the prover must generate and the veri£er accept at the completion of the execution. For reasons of brevity we do not discuss the CR2 setting much in this abstract, and refer the reader to the full version of this paper [3].

We clarify that the novel feature of our work is the consideration of reset attacks for identification. However our settings are de£ned in such a way that the traditional concurrent attacks as considered by [6, 10] and others are incorporated, so that security against these attacks is achieved by our protocols.

1.3 Four paradigms for identification secure against reset attack

As we explained above, the standard proof of knowledge based paradigm fails to provide identi£cation in the resettable setting. In that light, it may not be clear how to even prove the existence of a solution to the problem. Perhaps surprisingly however, not only can the existence of solutions be proven under the minimal assumption of a one-way function, but even simple and ef£cient solutions can be designed.

This is done in part by returning to some earlier paradigms. Zero-knowledge proofs of knowledge and identification are so strongly linked in contemporary cryptography that it is sometimes forgotten that these in fact replaced earlier identification techniques largely due to the ef£ciency gains they brought. In considering a new adversarial setting it is thus natural to £rst return to older paradigms and see whether they can be "lifted" to the resettable setting. We propose in particular signature and encryption based solutions for resettable identification and prove them secure in both the CR1 and the CR2 settings. We then present a general method for transforming identification protocols secure in a concurrent but non-reset setting to ones secure in a reset setting. Finally we return to the zero-knowledge ideas and provide a new paradigm based on zero-knowledge proofs of membership as opposed to proofs of knowledge.

SIGNATURE BASED IDENTIFICATION. The basic idea of the signature based paradigm is for Alice convinces Bob that she is Alice, by being "able to" sign random documents of Bob's choice. This is known (folklore) to yield a secure identi£cation scheme in the serial non-reset setting of [11] as long as the signature scheme is secure in the sense of [16]. It is also known to be secure in the concurrent non-reset setting [1]. But it fails in general to be secure in the resettable setting because an adversary can obtain signatures of different messages under the same prover coins. What we show is that the paradigm yields secure solutions in the resettable setting if certain special kinds of signature schemes are used. (The signing algorithm should be deterministic and stateless.) In the CR1 setting the basic protocol using such signature schemes suf£ces. The CR2 setting is more complex and we need to modify the protocol to include "challenges" sent by the prover. Since signature schemes with the desired properties exist (and even ef£cient ones exist) we obtain resettable identi£cation schemes proven secure under minimal assumptions for both the CR1 and the CR2 settings, and also obtain some ef£cient speci£c protocols.

ENCRYPTION BASED IDENTIFICATION. In the encryption based paradigm, Alice convinces Bob she is Alice, by being "able to" decrypt ciphertexts which Bob created. While the basic idea goes back to symmetric authentication techniques of the seventies, modern treatments of this paradigm appeared more recently in [9, 1, 10] but did not consider reset attacks. We show that under an appropriate condition on the encryption scheme —namely that it be secure against chosen-ciphertext attacks— a resettable identi£cation protocol can be obtained. As before the simple solution for the CR1 setting needs to be modi£ed before it will work in the CR2 setting.

TRANSFORMING STANDARD PROTOCOLS. Although Fiat-Shamir like identi£cation protocols are not secure in the context of reset attacks, with our third paradigm we show how to turn practical identi£cation schemes into secure ones in the CR1 and CR2 settings. The solution relies on the techniques introduced in [8] and utilizes pseudorandom functions and trapdoor commitments. It applies to most of the popular identi£cation schemes, like Fiat-Shamir [13], Okamoto-Schnorr [20, 18] or Okamoto-Guillou-Quisquater [17, 18].

ZK PROOF OF MEMBERSHIP BASED IDENTIFICATION. In the zero-knowledge proofs of membership paradigm, Alice convinces Bob she is Alice, by being "able to" prove membership in a hard language L, rather than by proving she has a witness for language

L. She does so by employing a resettable zero-knowledge proof of language membership for L as de£ned in [8] . Both Alice and Bob will need to have a public-key to enable the protocol. Alice's public-key de£nes who she is, and Bob's public-key enables him to verify her identity in a secure way. We adopt the general protocol for membership in NP languages of [8] for the purpose of identi£cation. The identi£cation protocols are constant round. What makes this work is the fact that the protocol for language membership ($x \in L$) being zero-knowledge implies "learning nothing" about x in a very strong sense — a veri£er cannot subsequently convince anyone else that $x \in L$ with non-negligible probability. We note that while we can make this approach work using resettable zero-knowledge proofs, it does not seem to work using resettable witness indistinguishable proofs for ID protocols.

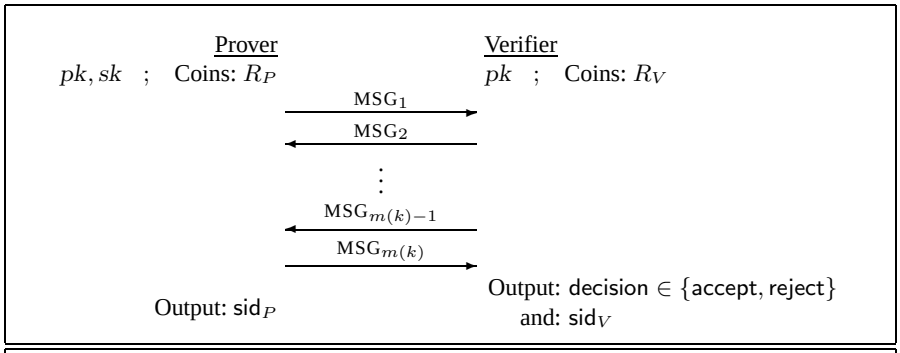
PERSPECTIVE. Various parts of the literature have motivated the study of zero-knowledge protocols secure against strong attacks such as concurrent or reset in part by the perceived need for such tools for the purpose of applications such as identification in similar attack settings. While the tools might be suf£cient for identification, they are not necessary. Our results demonstrate that identification is much easier than zero-knowledge and the latter is usually an overkill for the former.

2 De£nitions

If $A(\cdot,\cdot,\ldots)$ is a randomized algorithm then $y \leftarrow A(x_1, x_2, \ldots; R)$ means y is assigned the unique output of the algorithm on inputs $x_1, x_2, \ldots$ and coins R, while $y \leftarrow A(x_1, x_2, \ldots)$ is shorthand for £rst picking R at random (from the set of all strings of some appropriate length) and then setting $y \leftarrow A(x_1, x_2, \ldots; R)$. If $x_1, x_2, \ldots$ are strings then $x_1 \| x_2 \| \cdots$ denotes an encoding under which the constituent strings are uniquely recoverable. It is assumed any string x can be uniquely parsed as an encoding of some sequence of strings. The empty string is denoted ε.

An identi£cation protocol proceeds as depicted in Figure 1. The prover has a secret key sk whose matching public key pk is held by the veri£er. (In practice the prover might provide its public key, and the certi£cate of this public key, as part of the protocol, but this is better slipped under the rug in the model.) Each party computes its next message as a function of its keys, coins and the current conversation pre£x. The number of moves $m(k)$ is odd so that the £rst and last moves belong to the prover. (An identi£cation protocol is initiated by the prover who at the very least must provide a request to be identi£ed.) At the end of the protocol the veri£er outputs a decision to either accept or reject. Each party may also output a session id. (Sessions ids are relevant in the CR2 setting but can be ignored for the CR1 setting.) A particular protocol is described by a (single) *protocol description* function $\mathcal{ID}$ which speci£es how all associated processes —key generation, message computation, session id or decision computation— are implemented. (We say that $\mathcal{ID}$ is for the CR1 setting if $\mathsf{sid}_P = \mathsf{sid}_V = \varepsilon$, meaning no session ids are generated.) The second part of Figure 1 shows how it works: the £rst argument to $\mathcal{ID}$ is a keyword —one of keygen, prvmsg, vfmsg, prvsid, vfend— which invokes the subroutine responsible for that function on the other arguments.

Naturally, a correct execution of the protocol (meaning one in the absence of an adversary) should lead the veri£er to accept. To formalize this "completeness" require-



$(pk, sk) \leftarrow \mathcal{ID}(\mathsf{keygen}, k)$ —Randomized process to generate a public key pk and matching secret key sk
$\text{MSG}_{2j+1} \leftarrow \mathcal{ID}(\mathsf{prvmsg}, sk, \text{MSG}_1 \| \cdots \text{MSG}_{2j}; R_P)$ —$(1 \leq 2j+1 \leq m(k))$ Next prover message as a function of secret key, conversation pre£x and coins R_P
$\text{MSG}_{2j} \leftarrow \mathcal{ID}(\mathsf{vfmsg}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j-1}; R_V)$ —$(2 \leq 2j \leq m(k)-1)$ Next veri£er message as a function of public key, conversation pre£x and coins R_V
$\mathsf{sid}_P \leftarrow \mathcal{ID}(\mathsf{prvsid}, sk, \text{MSG}_1 \| \cdots \| \text{MSG}_{m(k)}; R_P)$ —Prover's session id as a function of secret key, full conversation and coins
$\mathsf{sid}_V \| \mathsf{decision} \leftarrow \mathcal{ID}(\mathsf{vfend}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{m(k)}; R_V)$ —Veri£er session id and decision (accept or reject) as a function of public key, full conversation and coins

Fig. 1. The prover sends the £rst and last messages in an $m(k)$-move identification protocol at the end of which the verifier outputs a decision and each party optionally outputs a session id. The *protocol description* function $\mathcal{ID}$ specifies all processes associated to the protocol.

ment we consider an *adversary-free execution* of the protocol $\mathcal{ID}$ which proceeds as described in the following experiment:

$(pk, sk) \leftarrow \mathcal{ID}(\mathsf{keygen}, k)$; Choose tapes R_P, R_V at random
$\text{MSG}_1 \leftarrow \mathcal{ID}(\mathsf{prvmsg}, sk, \varepsilon; R_P)$
`For` $j = 1$ `to` $\lfloor m(k)/2 \rfloor$ `do`
  $\text{MSG}_{2j} \leftarrow \mathcal{ID}(\mathsf{vfmsg}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j-1}; R_V)$
  $\text{MSG}_{2j+1} \leftarrow \mathcal{ID}(\mathsf{prvmsg}, sk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j}; R_P)$
`EndFor`
$\mathsf{sid}_P \leftarrow \mathcal{ID}(\mathsf{prvsid}, sk, \text{MSG}_1 \| \cdots \| \text{MSG}_{m(k)}; R_P)$
$\mathsf{sid}_V \| \mathsf{decision} \leftarrow \mathcal{ID}(\mathsf{vfend}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{m(k)}; R_V)$

The *completeness condition* is that, in the above experiment, the probability that $\mathsf{sid}_P = \mathsf{sid}_V$ and $\mathsf{decision} = \mathsf{accept}$ is 1. (The probability is over the coin tosses of $\mathcal{ID}(\mathsf{keygen}, k)$ and the random choices of R_P, R_V.) As always, the requirement can be relaxed to only ask for a probability close to one.

Fix an identi£cation protocol description function $\mathcal{ID}$ and an adversary I. Associated to them is $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$, depicted in Figure 2, which is used to de£ne the

Experiment $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$ —Execution of protocol $\mathcal{ID}$ with adversary I and security parameter k in the CR1 setting

Initialization:

(1) $(pk, sk) \leftarrow \mathcal{ID}(\mathsf{keygen}, k)$ / Pick keys via randomized key generation algorithm /

(2) Choose tape R_V for verifier at random ; $C_V \leftarrow 0$ / Coins and message counter for verifier /

(3) $p \leftarrow 0$ / Number of active prover instances /

Execute adversary I on input pk and reply to its oracle queries as follows:

- When I makes query WakeNewProver / Activate a new prover instance /
  (1) $p \leftarrow p + 1$; Pick a tape R_p at random ; Return p
- When I makes query $\mathsf{Send}(\mathsf{prvmsg}, i, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j})$ with $0 \leq 2j < m(k)$ and $1 \leq i \leq p$
  (1) If $C_V \neq 0$ then Return $\perp$ / Interaction with prover instance allowed only before interaction with verifier begins /
  (2) $\text{MSG}_{2j+1} \leftarrow \mathcal{ID}(\mathsf{prvmsg}, sk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j}; R_i)$
  (3) Return MSG_{2j+1}
- When I makes query $\mathsf{Send}(\mathsf{vfmsg}, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j-1})$ with $1 \leq 2j - 1 \leq m(k)$
  (1) $C_V \leftarrow C_V + 2$
  (2) If $2j < C_V$ then Return $\perp$ / Not allowed to reset the verifier /
  (3) If $2j - 1 < m(k) - 1$ then $\text{MSG}_{2j} \leftarrow \mathcal{ID}(\mathsf{vfmsg}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j-1}; R_V)$; Return MSG_{2j}
  (4) If $2j - 1 = m(k)$ then decision $\leftarrow \mathcal{ID}(\mathsf{vfend}, pk, \text{MSG}_1 \| \cdots \| \text{MSG}_{2j}; R_V)$
  (5) Return decision

Did I win? When I has terminated set $\text{WIN}_I = \mathsf{true}$ if $\mathsf{decision} = \mathsf{accept}$.

Fig. 2. Experiment describing execution of identi£cation protocol $\mathcal{ID}$ with adversary I and security parameter k in the CR1 setting.

security of $\mathcal{ID}$ in the CR1 setting. (In this context it is understood that $\mathcal{ID}$ is for the CR1 setting, meaning does not produce session ids.) The experiment gives the adversary appropriate access to prover instance oracles $\mathsf{Prover}^1, \mathsf{Prover}^2, \ldots$ and a single veri£er oracle, let it query these subject to certain restrictions imposed by the experiment, and then determine whether it "wins". The interface to the prover instance oracles and the veri£er oracle (which, in the experiment, are implicit, never appearing by name) is via oracle queries; the experiment enumerates the types of queries and shows how answers are provided to them.

The experiment begins with some initializations which include choosing of the keys. Then the adversary is invoked on input the public key. A WakeNewProver query activates a new prover instance Prover^p by picking a random tape R_p for it. (A random tape for a prover instance is chosen exactly once and all messages of this prover instance are

then computed with respect to this tape. The tape of a speci£c prover instance cannot be changed, or "reset", once chosen.) A $\mathsf{Send}(\mathsf{prvmsg}, i, x)$ query —viewed as sent to prover instance Prover^i— results in the adversary being returned the next prover message computed as $\mathcal{ID}(\mathsf{prvmsg}, sk, x; R_i)$. (It is assumed that $x = \text{MSG}_1 \| \cdots \| \text{MSG}_{2j}$ is a *valid conversation pre£x*, meaning contains an even number of messages $2j < m(k)$, else the query is not valid.) Resetting is captured by allowing arbitrary (valid) conversation pre£xes to be queried. (For example the adversary might try $\text{MSG}_1 \| \text{MSG}_2$ for many different values of MSG_2, corresponding to successively resetting the prover instance to the point where it receives the second protocol move.) Concurrency is captured by the fact that any activated prover instances can be queried.

A $\mathsf{Send}(\mathsf{vfmsg}, x)$ query is used to invoke the veri£er on a conversation pre£x x and results in the adversary being returned either the next veri£er message computed as $\mathcal{ID}(\mathsf{vfmsg}, pk, x; R_V)$ —this when the veri£er still has a move to make— or the decision computed as $\mathcal{ID}(\mathsf{vfend}, pk, x; R_V)$ —this when x corresponds to a full conversation. (Here R_V was chosen at random in the experiment initialization step. It is assumed that $x = \text{MSG}_1 \| \cdots \| \text{MSG}_{2j-1}$ is a valid conversation pre£x, meaning contains an odd number of messages $1 \leq 2j - 1 \leq m(k)$, else the query is not valid.) Unlike a prover instance, resetting the (single) veri£er instance is not allowed. (Our signature and encryption based protocols are actually secure even if veri£er resets are allowed, but since the practical need to consider this attack is not apparent, the de£nition excludes it.) This is enforced explicitly in the experiments via the veri£er message counter C_V.

In the CR1 setting, the adversary's actions are divided into two phases. In the £rst phase it interacts with the prover instances, not being allowed to interact with the veri£er; in the second phase it is denied access to the prover instances and tries to convince the veri£er to accept. $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$ enforces this by returning $\perp$ in reply to a $\mathsf{Send}(\mathsf{prvmsg}, i, x)$ unless $C_V = 0$.

The adversary wins if it makes the verifier instance accept. The parameter WIN_I is set accordingly in $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$. The de£nition of the protocol is responsible for ensuring that both parties reject a received conversation pre£x if it is inconsistent with their coins. It is also assumed that the adversary never repeats an oracle query. We can now provide de£nitions of security for protocol $\mathcal{ID}$.

De£nition 1. [Security of an ID protocol in the CR1 setting] *Let $\mathcal{ID}$ be an identi£cation protocol description for the* CR1 *setting. Let I be an adversary (called an impersonator in this context) and let k be the security parameter. The advantage of impersonator I is*

$$\mathbf{Adv}^{\text{id-cr1}}_{\mathcal{ID},I}(k) = \Pr[\,\text{WIN}_I = \mathsf{true}\,]$$

where the probability is with respect to $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$. Protocol $\mathcal{ID}$ is said to be polynomially-secure in the CR1 *setting if $\mathbf{Adv}^{\text{id-cr1}}_{\mathcal{ID}}(\cdot)$ is negligible for any impersonator I of time-complexity polynomial in k.* ∎

We adopt the convention that the *time-complexity* $t(k)$ of an adversary I is the execution time of the entire experiment $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$, including the time taken for initialization, computation of replies to adversary oracle queries, and computation of

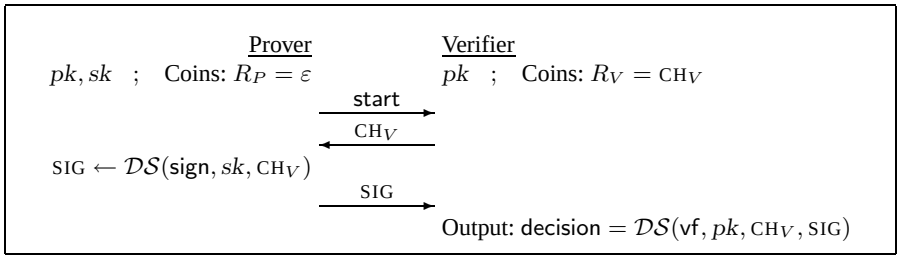


Fig. 3. Reset-secure identification protocol $\mathcal{ID}$ for the CR1 setting based on a deterministic, stateless digital signature scheme $\mathcal{DS}$.

WIN_I. We also de£ne the *query-complexity* $q(k)$ of I as the number of $\mathsf{Send}(\mathsf{prvmsg}, \cdot, \cdot)$ queries made by I in $\mathbf{Experiment}^{\text{id-cr1}}_{\mathcal{ID},I}(k)$. It is always the case that $q(k) \leq t(k)$ so an adversary of polynomial time-complexity has polynomial query-complexity. These de£nitions and conventions can be ignored if polynomial-security is the only concern, but simplify concrete security considerations to which we will pay some attention later.

A de£nition of security for the CR2 setting can be found in [3].

3 CR1-secure Identification protocols

Four paradigms are illustrated: signature based, encryption based, identification based, and zero-knowledge based.

3.1 A signature based protocol

We assume knowledge of background in digital signatures as summarized in [3].

SIGNATURE BASED IDENTIFICATION. A natural identi£cation protocol is for the veri£er to issue a random challenge CH_V and the prover respond with a signature of CH_V computed under its secret key sk. (Pre£x the protocol with an initial start move by the prover to request start of an identi£cation process, and you have a three move protocol.) This simple protocol can be proven secure in the serial, non-resettable (ie. standard smartcard) setting of [11] as long as the signature scheme meets the notion of security of [16] . (This result seems to be folklore.) The same protocol has also been proven to provide authentication in the concurrent, non-resettable (ie. standard network) setting [1]. (The intuition in both cases is that the only thing an adversary can do with a prover oracle is feed it challenge strings and obtain their signatures, and if the scheme is secure against chosen-message attack this will not help the adversary forge a signature of a challenge issued by the veri£er unless it guesses the latter, and the probability of the last event can be made small by using a long enough challenge.) This protocol is thus a natural candidate for identi£cation in the resettable setting.

However this protocol does not always provide security in the resettable setting. The intuition described above breaks down because resetting allows an adversary to obtain the signatures of different messages under the same set of coins. (It can activate a

prover instance and then query it repeatedly with different challenges, thereby obtaining their signatures with respect to a £xed set of coin tosses.) As explained in [3], this is not covered by the usual notion of a chosen-message attack used to de£ne security of signature schemes in [16]. And indeed, for many signature schemes it is possible to forge the signature of a new message if one is able to obtain the signatures of several messages under one set of coins. Similarly, if the signing algorithm is stateful, resetting allows an adversary to make the prover release several signatures computed using one value of the state variable —effectively, the prover does not get a chance to update its state is it expects to— again leading to the possibility of forgery on a scheme secure in the standard sense.

The solution is simple: restrict the signature scheme to be stateless and deterministic. In [3] we explain how signatures schemes can be imbued with these attributes so that stateless, deterministic signature schemes are available.

PROTOCOL AND SECURITY. Let $\mathcal{DS}$ be a deterministic, stateless signature scheme. Figure 3 illustrates the ¤ows of the associated identification protocol $\mathcal{ID}$. A parameter of the protocol is the length $vcl(k)$ of the verifier's random challenge. The prover is deterministic and has random tape ε while the verifier's random tape is CH_V. Refer to De£nition 1 and [3] for the meanings of terms used in the theorem below, and to [3] for the proof.

Theorem 1. [Concrete security of the signature based ID scheme in the CR1 **setting]** *Let $\mathcal{DS}$ be a deterministic, stateless signature scheme, let $vcl(\cdot)$ be a polynomially-bounded function, and let $\mathcal{ID}$ be the associated identi£cation scheme as per Figure 3. If I is an adversary of time-complexity $t(\cdot)$ and query-complexity $q(\cdot)$ attacking $\mathcal{ID}$ in the* CR1 *setting then there exists a forger F attacking $\mathcal{DS}$ such that*

$$\mathbf{Adv}^{\text{id-cr1}}_{\mathcal{ID},I}(k) \leq \mathbf{Adv}^{\text{ds}}_{\mathcal{DS},F}(k) + \frac{q(k)}{2^{vcl(k)}} \ . \tag{1}$$

Furthermore F has time-complexity $t(k)$ and makes at most $q(k)$ signing queries in its chosen-message attack on $\mathcal{DS}$. ∎

This immediately implies the following:

Corollary 1. [Polynomial-security of the signature based ID scheme in the CR1 **setting]** *Let $\mathcal{DS}$ be a deterministic, stateless signature scheme, let $vcl(k) = k$, and let $\mathcal{ID}$ be the associated identification scheme as per Figure 3. If $\mathcal{DS}$ is polynomially-secure then $\mathcal{ID}$ is polynomially-secure in the* CR1 *setting.* ∎

We show in [3] that this implies:

Corollary 2. [Existence of an ID scheme polynomially-secure in the CR1 **setting]** *Assume there exists a one-way function. Then there exists an identification scheme that is polynomially-secure in the* CR1 *setting.*

3.2 An encryption based protocol

ENCRYPTION BASED IDENTIFICATION. The idea is simple: the prover proves its identity by proving its ability to decrypt a ciphertext sent by the veri£er. This basic idea goes

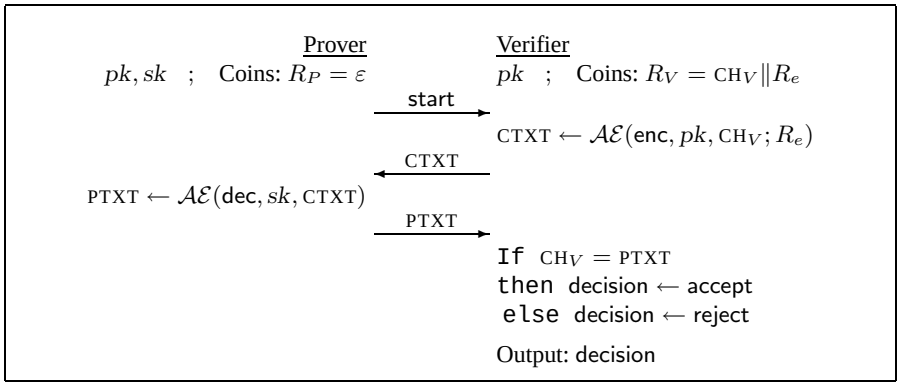


Fig. 4. Reset-secure identification protocol $\mathcal{ID}$ for the CR1 setting based on a chosen-ciphertext attack secure asymmetric encryption scheme $\mathcal{AE}$.

back to early work in entity authentication where the encryption was usually symmetric (ie. private-key based). These early protocols however had no supporting de£nitions or analysis. The £rst "modern" treatment is that of [9] who considered the paradigm with regard to providing deniable authentication and identi£ed non-malleability under chosen-ciphertext attack —equivalently, indistinguishability under chosen-ciphertext attack [2, 9]— as the security property required of the encryption scheme. Results of [1, 10, 9] imply that the protocol is a secure identi£cation scheme in the concurrent non-reset setting, but reset attacks have not been considered before.

PROTOCOL AND SECURITY. Let $\mathcal{AE}$ be an asymmetric encryption scheme polynomially-secure against chosen-ciphertext attack. Figure 4 illustrates the ¤ows of the associated identi£cation protocol $\mathcal{ID}$. A parameter of this protocol is the length $vcl(k)$ of the veri£er's random challenge. The veri£er sends the prover a ciphertext formed by encrypting a random challenge, and the prover identi£es itself by correctly decrypting this to send the veri£er back the challenge. The prover is deterministic, having random tape ε. We make the coins R_e used by the encryption algorithm explicit, so that the veri£er's random tape consists of the challenge —a random string of length $vcl(k)$ where vcl is a parameter of the protocol— and coins suf£cient for one invocation of the encryption algorithm. Refer to De£nition 1 and [3] for the meanings of terms used in the theorem below, and to [3] for the proof.

Theorem 2. [Concrete security of the encryption based ID scheme in the CR1 setting] *Let $\mathcal{AE}$ be an asymmetric encryption scheme, let $vcl(\cdot)$ a polynomially-bounded function, and let $\mathcal{ID}$ be the associated identi£cation scheme as per Figure 4. If I is an adversary of time-complexity $t(\cdot)$ and query-complexity $q(\cdot)$ attacking $\mathcal{ID}$ in the CR1 setting then there exists an eavesdropper E attacking $\mathcal{AE}$ such that*

$$\mathbf{Adv}^{\text{id-cr1}}_{\mathcal{ID},I}(k) \leq \mathbf{Adv}^{\text{lr-cca}}_{\mathcal{AE},E}(k) + \frac{2q(k)+2}{2^{vcl(k)}} \ . \tag{2}$$

Furthermore E has time-complexity $t(k)$, makes one query to its lr-encryption oracle, and at most $q(k)$ queries to its decryption oracle. ∎

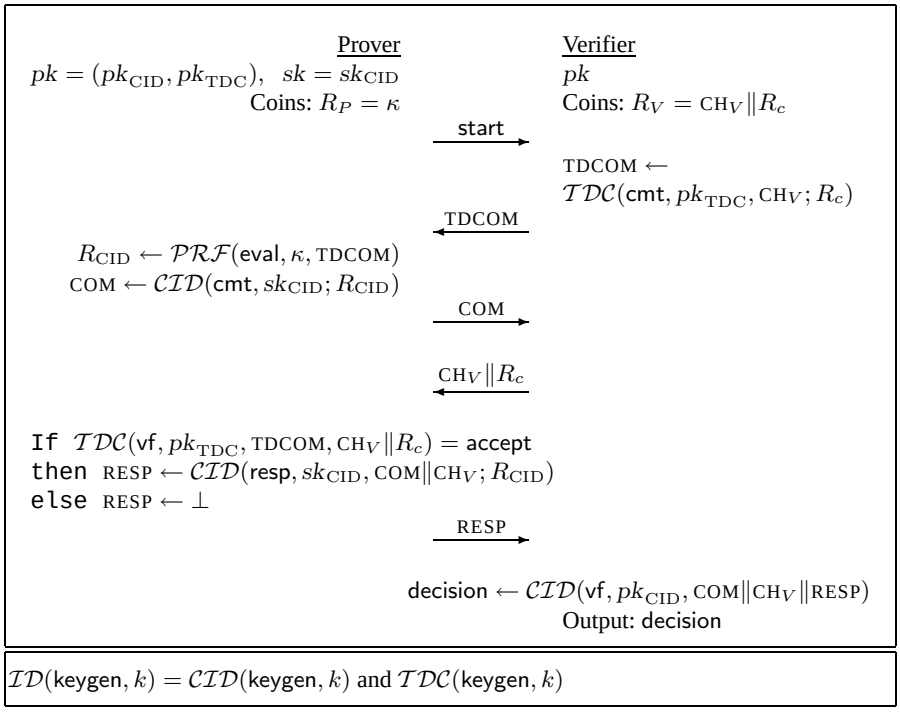


Fig. 5. Reset-secure identification protocol $\mathcal{ID}$ for the CR1 setting based on an identification scheme $\mathcal{CID}$ secure against non-resetting CR1 attacks

This immediately implies the following:

Corollary 3. [Polynomial-security of the encryption based ID scheme in the CR1 **setting]** *Let $\mathcal{AE}$ be an asymmetric encryption scheme, let $vcl(k) = k$, and let $\mathcal{ID}$ be the associated identification scheme as per Figure 4. If $\mathcal{AE}$ is polynomially-secure against chosen-ciphertext attack then $\mathcal{ID}$ is polynomially-secure in the* CR1 *setting.* ∎

3.3 An identification based protocol

IDENTIFICATION BASED PROTOCOL. As discussed in the introduction, proof of knowledge based identi£cation protocols of the Fiat-Shamir type cannot be secure against reset attacks. In this section, however, we present a general transformation of such identi£cation schemes into secure ones in the CR1 setting. We start with identi£cation schemes that consists of three moves, an initial commitment COM of the prover, a random value CH_V, the challenge, of the veri£er and a conclusive response RESP from the prover. We call a protocol obeying this structure a canonical identi£cation scheme.

Loosely speaking, we will assume that the underlying canonical identical scheme $\mathcal{CID}$ is secure against non-resetting attacks in the CR1 model, i.e., against attacks where the adversary merely runs concurrent sessions with the prover without resets before

engaging in a verification. In addition to the Fiat-Shamir system [13], most of the well-known practical identification schemes also achieve this security level, for example Ong-Schnorr [19, 21] for some system parameters, Okamoto-Guillou-Quisquater [17, 18] and Okamoto-Schnorr [20, 18]. Nonetheless, there are also protocols which are only known to be secure against sequential attacks (e.g. [22]).

To avoid confusion with the derived scheme $\mathcal{ID}$, instead of writing $\mathsf{Send}(\mathsf{prvmsg},\ldots)$ and $\mathsf{Send}(\mathsf{vfmsg},\ldots)$, we denote the algorithms generating the commitment, challenge and response message for the CID-protocol $\mathcal{CID}$ by $\mathcal{CID}(\mathsf{cmt},\ldots)$, $\mathcal{CID}(\mathsf{chall},\ldots)$, and $\mathcal{CID}(\mathsf{resp},\ldots)$, respectively, and the veri£cation step by $\mathcal{CID}(\mathsf{vf},\ldots)$. We also write $\mathbf{Adv}^{\text{id-nr-cr1}}_{\mathcal{CID},I_{\mathrm{CID}}}(k)$ for the probability that an impersonator I_{CID} succeeds in an attack on scheme $\mathcal{CID}$ in the non-resetting CR1 setting.

PROTOCOL AND SECURITY. Our solution originates from the work of [8] about resettable zero-knowledge. In order to ensure that the adversary does not gain any advantage from resetting the prover, we insert a new £rst round into the CID-identification protocol in which the veri£er non-interactively commits to his challenge CH_V. The parameters for this commitment scheme become part of the public key. This keeps the adversary from resetting the prover to the challenge-message and completing the protocol with different challenges.

In addition, we let the prover determine the random values in his identi£cation by applying a pseudorandom function to the veri£er's initial commitment. Now, if the adversary resets the prover (with the same random tape) to the outset of the protocol and commits to a different challenge then the prover uses virtually independent randomness for this execution, although having the same random tape. On the other hand, using pseudorandom values instead of truly random coins does not weaken the original identi£cation protocol noticeably. Essentially, this prunes the CR1 adversary into a non-resetting one concerning executions with the prover.

In order to handle the intrusion try we use use a special, so-called trapdoor commitment scheme $\mathcal{TDC}$ for the veri£er's initial commitment. This means that there is a secret information such that knowledge of this secret allows to generate a dummy commitment and to £nd a valid opening to any value later on. Furthermore, the dummy commitment and the fake decommitment are identically distributed to an honestly given commitment and opening to the same value. Without knowing the secret a commitment is still solidly binding. Trapdoor commitment schemes exist under standard assumptions like the intractability of the discrete-log or the RSA or factoring assumption [7] and thus under the same assumptions that the aforementioned CID-identification protocols rely on.

Basically, a trapdoor commitment enables us to reduce an intrusion try of an impersonator I in the derived scheme $\mathcal{ID}$ to one for the CID-protocol. If I initiates a session with the veri£er in $\mathcal{ID}$ then we can £rst commit to a dummy value $0^{vcl(k)}$ without having to communicate with the veri£er in $\mathcal{CID}$. When I then takes the next step by sending COM, we forward this commitment to our veri£er in $\mathcal{CID}$ and learn the veri£er's challenge. Knowing the secret key sk_{TDC} for the trapdoor scheme we can then £nd a valid opening for our dummy commitment with respect to the challenge. Finally, we forward I's response in our attack.

The scheme is displayed in Figure 5. See [3] for de£nitions and notions. The discussion above indicates that any adversary I for $\mathcal{ID}$ does not have much more power than a non-resetting impersonator attacking $\mathcal{CID}$ and security of $\mathcal{ID}$ follows from the security of $\mathcal{CID}$.

Theorem 3. [Concrete security of the identi£cation based scheme in the CR1 **setting]** *Let $\mathcal{CID}$ be an* CID*-identification protocol and let $vcl(\cdot)$ be a polynomially-bounded function. Also, let $\mathcal{PRF}$ be a pseudorandom function family and denote by $\mathcal{TDC}$ a trapdoor commitment scheme. Let $\mathcal{ID}$ be the associated identi£cation scheme as per Figure 5. If I is an adversary of time-complexity $t(\cdot)$ and query-complexity $q(\cdot)$ attacking $\mathcal{ID}$ in the* CR1 *setting then there exists an adversary I_{CID} attacking $\mathcal{CID}$ in a non-resetting* CR1 *attack such that*

$$\mathbf{Adv}^{\text{id-cr1}}_{\mathcal{ID},I}(k) \leq q(k) \cdot \mathbf{Adv}^{\mathcal{PRF}}_{(t,q)}(k) + \mathbf{Adv}^{\mathcal{TDC}}_{t}(k) + \mathbf{Adv}^{\text{id-nr-cr1}}_{\mathcal{CID},I_{\mathrm{CID}}}(k)\ . \qquad (3)$$

Furthermore I_{CID} has time-complexity $t(k)$ and runs at most $q(k)$ sessions with the prover before trying to intrude.

As usual we have:

Corollary 4. [Polynomial-security of the identi£cation based scheme in the CR1 **setting]** *Let $\mathcal{PRF}$ be a polynomially-secure pseudorandom function family and let $\mathcal{TDC}$ be a polynomially-secure trapdoor commitment scheme, set $vcl(k) = k$, and let $\mathcal{ID}$ be the associated identi£cation scheme as per Figure 5. If $\mathcal{CID}$ is a polynomially-secure* CID*-identification protocol in the non-resetting* CR1 *model then $\mathcal{ID}$ is polynomially-secure in the* CR1 *setting.* ∎

Note that the public key in our CR1-secure identi£cation scheme consists of two independent parts, pk_{CID} and pk_{TDC}. For concrete schemes the key generation may be combined and simpli£ed. For instance, for Okamoto-Schnorr the public key of the identi£cation protocol describes a group of prime order q, two generators g_1, g_2 of that group and the public key $X = g_1^{x_1} g_2^{x_2}$ for secret $x_1, x_2 \in \mathsf{Z}_q$. The prover sends $\mathrm{COM} = g_1^{r_1} g_2^{r_2}$ and replies to the challenge CH_V by transmitting $y_i = r_i + \mathrm{CH}_V x_i \bmod q$ for $i = 1, 2$. In this case, the public key for the trapdoor commitment scheme could be given by $g_1, g_3 = g_1^z$ for random trapdoor $z \in \mathsf{Z}_q$, and the commitment function maps a value c and randomness R_c to $g_1^c g_3^{R_c}$.

3.4 A zero-knowledge based protocol

As we discussed in the Introduction the idea of [11] of proving identity by employing a zero knowledge proof of knowledge has been the accepted paradigm for identi£cation protocols in the smartcard setting. Unfortunately, as we indicated, in the resettable setting this paradigm cannot work.

RESETTABLE ZERO KNOWLEDGE BASED IDENTITY. We thus instead propose the following paradigm. Let L be a hard NP language for which there is no known ef£cient procedures for membership testing but for which there exists a randomized generating algorithm G which outputs pairs (x, w), where $x \in L$ and w is an NP-witness that $x \in L$. (The distribution according to which (x, w) is generated should be one for

which it is hard to tell whether $x \in L$ or not). Each user Alice will run G to get a pair (x, w) and will then publish x as its public key. To prove her identity Alice will run a resettable zero-knowledge proof that $x \in L$.

PROTOCOL. To implement the above idea we need resettable zero-knowledge proofs for L. For this we turn to the work of [8]. In [8] two resettable zero-knowledge proofs for any NP language are proposed: one which takes a non-constant number of rounds and works against a computationally unbounded prover, and one which only takes a constant number of rounds and works against computationally bounded provers (i.e argument) and requires the verifiers to have published public-keys which the prover can access. We propose to utilize the latter, for ef£ciency sake. Thus, to implement the paradigm, we require both prover and verifier to have public-keys accessible by each other. Whereas the prover's public key is x whose membership in L it will prove to the verifier, the verifier's public key in [8] is used for specifying a perfectly private computationally binding commitment scheme which the prover must use during the protocol. (Such commitment schemes exist based for example on the strong hardness of Discrete Log Assumption.)

SECURITY. We brie¤y outline how to prove that the resulting ID protocol is secure in the CR1 setting. Suppose not, and that after launching a CR1 attack, an imposter can now falsely identify himself with a non-negligible probability. Then, we will construct a polynomial time algorithm A to decide membership in L. On input x, A £rst launches the off-line resetting attack using x as the public key and the simulator – which exists by the zero-knowledge property – to obtain views of the protocol execution. (This requires that the simulator be black-box, but this is true in the known protocols.) If $x \in L$, this view should be identical to the view obtained during the real execution, in which case a successful attack will result, which is essentially a way for A to £nd a language membership proof. If x not in L, then by the soundness property of a zero-knowledge proof, no matter what the simulator outputs, it will not be possible to prove membership in L.

Acknowledgments

The second author thanks Ran Canetti for discussions about resettable security. The £rst author is supported in part by a 1996 Packard Foundation Fellowship in Science and Engineering.

References

1. M. BELLARE, R. CANETTI, AND H. KRAWCZYK, "A modular approach to the design and analysis of authentication and key exchange protocols," *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, 1998.
2. M. BELLARE, A. DESAI, D. POINTCHEVAL AND P. ROGAWAY, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

3. M. BELLARE, M. FISCHLIN, S. GOLDWASSER AND S. MICALI, "Identification protocols secure against reset attacks," Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir`.
4. M. BELLARE AND O. GOLDREICH, "On de£ning proofs of knowledge," *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.
5. M. BELLARE, D. POINTCHEVAL AND P. ROGAWAY, "Authenticated key exchange secure against dictionary attack," *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.
6. M. BELLARE AND P. ROGAWAY, "Entity authentication and key distribution", *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science Vol. 773, D. Stinson ed., Springer-Verlag, 1993.
7. G. BRASSARD, D. CHAUM AND C. CRÉPEAU, "Minimum Disclosure Proofs of Knowledge," *Journal of Computer and Systems Science*, Vol. 37, No. 2, 1988, pp. 156–189.
8. R. CANETTI, S. GOLDWASSER, O. GOLDREICH AND S. MICALI, "Resettable zero-knowledge," *Proceedings of the 32nd Annual Symposium on the Theory of Computing*, ACM, 2000.
9. D. DOLEV, C. DWORK AND M. NAOR, "Non-malleable cryptography", *SIAM J. on Computing*, 2001. Preliminary version in STOC 91.
10. C. DWORK, M. NAOR AND A. SAHAI, "Concurrent zero-knowledge," *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, 1998.
11. U. FEIGE, A. FIAT AND A. SHAMIR, "Zero-knowledge proofs of identity," *J. of Cryptology*, Vol. 1, 1988, pp. 77-94.
12. U. FEIGE AND A. SHAMIR, "Witness indistinguishable and witness hiding protocols," *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.
13. A. FIAT AND A. SHAMIR, "How to prove yourself: Practical solutions to identification and signature problems," *Advances in Cryptology – CRYPTO '86*, Lecture Notes in Computer Science Vol. 263, A. Odlyzko ed., Springer-Verlag, 1986.
14. O. GOLDREICH, S. GOLDWASSER AND S. MICALI, "How to construct random functions," *Journal of the ACM*, Vol. 33, No. 4, 1986, pp. 210–217.
15. S. GOLDWASSER, S. MICALI AND C. RACKOFF, "The knowledge complexity of interactive proof systems," *SIAM J. on Computing*, Vol. 18, No. 1, pp. 186–208, February 1989.
16. S. GOLDWASSER, S. MICALI AND R. RIVEST, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing*, Vol. 17, No. 2, April 1988, pp. 281–308.
17. L.C. GUILLOU AND J.-J. QUISQUATER, "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing Both Transmission and Memory," *Advances in Cryptology – EUROCRYPT '88*, Lecture Notes in Computer Science Vol. 330, C. Gunther ed., Springer-Verlag, 1988.
18. T. OKAMOTO, "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes," *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.
19. H. ONG AND C.P. SCHNORR, "Fast Signature Generation with a Fiat-Shamir Identification Scheme" *Advances in Cryptology – EUROCRYPT '90*, Lecture Notes in Computer Science Vol. 473, I. Damgård ed., Springer-Verlag, 1990.
20. C.P. SCHNORR, "Eff£cient Signature Generation by Smart Cards," *J. of Cryptology*, Vol. 4, 1991, pp. 161–174.
21. C.P. SCHNORR, "Security of 2^t-Root Identi£cation and Signatures" *Advances in Cryptology – CRYPTO '96*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

22. V. SHOUP, "On the Security of a Practical Identification Scheme," *J. of Cryptology*, Vol. 12, 1999, pp. 247–260.