GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks

Mihir Bellare and Adriana Palacio

Dept. of Computer Science & Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA.
Email: {mihir,apalacio}@cs.ucsd.edu
URL: http://www-cse.ucsd.edu/users/{mihir,apalacio}

Abstract. The Guillou-Quisquater (GQ) and Schnorr identification schemes are amongst the most efficient and best-known Fiat-Shamir follow-ons, but the question of whether they can be proven secure against impersonation under active attack has remained open. This paper provides such a proof for GQ based on the assumed security of RSA under one more inversion, an extension of the usual one-wayness assumption that was introduced in [5]. It also provides such a proof for the Schnorr scheme based on a corresponding discrete-log related assumption. These are the first security proofs for these schemes under assumptions related to the underlying one-way functions. Both results extend to establish security against impersonation under concurrent attack.

1 Introduction

The Guillou-Quisquater (GQ) [20] and Schnorr [26] identification schemes are amongst the most efficient and best known Fiat-Shamir [16] follow-ons, but the question of whether they can be proved secure against impersonation under active attack has remained open. This paper addresses this question, as well as its extension to even stronger attacks, namely concurrent ones. We begin with some background.

1.1 Identification schemes and their security

An identification (ID) scheme enables a prover holding a secret key to identify itself to a verifier holding the corresponding public key. Fiat and Shamir (FS) [16] showed how the use of zero-knowledge techniques [19] in this area could lead to efficient schemes, paving the road for numerous successors including [20, 26], which are comparable to FS in computational cost but have much smaller key sizes.

The accepted framework for security notions for identification schemes is that of Feige, Fiat and Shamir [14]. As usual, one considers adversary goals as well as adversary capabilities, or attacks. The adversary goal is impersonation: playing the role of prover but denied the secret key, it should have negligible probability of making the verifier accept. Towards this goal, one can allow it various attacks on the honest, secret-key equipped prover which, as per [14], take place and complete before the impersonation attempt. The weakest reasonable attack is a passive attack, in which the adversary obtains transcripts of interactions between the prover and verifier. However, the attack suggested by [14] as defining the main notion of security is an active attack in which the adversary plays the role of cheating verifier, interacting with the prover numerous times before the impersonation attempt.

Security against impersonation under active attack has been the classical goal of identification schemes. However, interest has been growing in stronger attacks, namely concurrent ones. Here, the adversary would still play the role of cheating verifier prior to impersonation, but could interact with many different prover "clones" concurrently. The clones all have the same secret key but are initialized with independent coins and maintain their own state. Security against impersonation under concurrent attack implies security against impersonation under active attack.

Analyses often approach the establishment of security against impersonation via consideration of whether or not the protocol is a proof of knowledge, honest-verifier zero knowledge, witness indistinguishable [15] and so on. These auxiliary properties are important and useful tools, but not the end goal, which remains establishing security against impersonation.

1.2 The GQ scheme and our results about it

GQ is RSA based. The prover's public key is (N, e, X), where N is an RSA modulus, e is a prime RSA exponent, and $X \equiv x^e \pmod N$ where $x \in \mathbb{Z}_N^*$ is the prover's secret key. As typical for practical ID schemes, the protocol, depicted in Figure 2, has three moves: the prover sends a "commitment," the verifier sends a random challenge, the prover sends a "response," and the verifier then accepts or rejects. The protocol is honest-verifier zero knowledge and a proof of knowledge of x [20], and it follows easily that it is secure against impersonation under passive attack, assuming RSA is one-way.

The main question is whether the protocol is secure against impersonation under active attack. No attack has been found. However, no proof of security has been provided either. Furthermore, it is difficult to imagine such a proof being based solely on the assumption that RSA is one-way. (The prover response is the RSA inverse of a point that is a function of the verifier challenge, giving a cheating verifier some sort of limited chosen-ciphertext attack capability, something one-wayness does not consider.) In other words, the protocol seems to be secure against impersonation under active attack, but due to properties of RSA that go beyond mere one-wayness.

The research community is well aware that RSA has important strengths beyond one-wayness, and have captured some of them with novel assumptions. Examples include the strong RSA assumption, introduced in [17, 2] and exploited in [18, 13]; the dependent-RSA assumptions [24]; and the assumption of security under one more inversion [5]. The intent, or hope, of introducing such assumptions is that they underlie not one but numerous uses or protocols. Thus our approach is to attempt to build on this existing experience, and prove security based on one of these assumptions.

We prove that the GQ identification scheme is secure against impersonation, under both active and concurrent attacks, under the assumption that RSA is secure under one more inversion. The precise statement of the result is Corollary 1. Let us now explain the assumption.

Security of RSA under one more inversion, as introduced in [5], considers an adversary given input an RSA public key N, e, and access to two oracles. The *challenge oracle* takes no inputs and returns a random target point in $\mathbb{Z}_N^*$, chosen anew each time the oracle is invoked. The *inversion oracle* given $y \in \mathbb{Z}_N^*$ returns $y^d \bmod N$, where d is the decryption exponent corresponding to e. The assumption states that it is computationally infeasible for the adversary to output correct inverses of all the target points if the number of queries it makes to its inversion oracle is strictly less than the number of queries it makes to its challenge oracle. (When the adversary makes one challenge query and no inversion queries, this is the standard one-wayness assumption, which is why security under one more inversion is considered an extension of the standard one-wayness assumption.) This assumption was used in [5] to prove the security of Chaum's RSA-based blind-signature scheme [12] in the random oracle model. (Our results, however, do not involve random oracles.) It was also used in [6] to prove the security of an RSA-based transitive signature scheme due to [21].

Our result is based on a relatively novel and strong assumption that should be treated with caution. But the result still has value. It reduces the security of the GQ identification scheme to a question which is solely about the security of the RSA function. Cryptanalysts need no longer attempt to attack the identification scheme, but can instead concentrate on a simply stated assumption about RSA, freeing themselves from the details of the identification model. Furthermore, our result helps clarify and unify the global picture of protocol security by showing that the properties of RSA underlying the security of the GQ identification scheme and Chaum's RSA-based blind-signature scheme are the same. Thus our result brings the benefit we usually expect with a proof of security, namely reduction of the security of many cryptographic problems to a single number-theoretic problem. Finally, a proof under a stronger than standard assumption is better than no proof at all in the context of a problem whose provable security has remained an open question for more than ten years.

1.3 The Schnorr scheme and our results about it

The Schnorr identification scheme is discrete logarithm based. The prover's public key is (g, X), where g is a generator of a suitable prime-order group and $X = g^x$ where x is the prover's secret key. The protocol, having the usual three-move format, is depicted in Figure 4. Again the protocol is honest-verifier zero knowledge and a proof of knowledge of x [26], and it follows easily that it is secure against impersonation under passive attack, assuming hardness of computation of discrete logarithms in the underlying group. (That is, one-wayness of the discrete exponentiation function.) As with GQ, the scheme appears to be secure against impersonation under active attack in the sense that no attacks are known, but proving security has remained open.

We prove that the Schnorr scheme is secure against impersonation, under both active and concurrent attacks, under the assumption that discrete exponentiation is secure under one more inversion in the underlying group. The precise statement of the result is Corollary 2. The assumption, also introduced in [5], is the natural analogue of the one we used for RSA. The adversary gets input the generator g. Its challenge oracle returns a random group element, and its inversion oracle computes discrete logarithms relative to g. The assumption states that it is computationally infeasible for the adversary to output correct discrete logarithms of all the target points if the number of queries it makes to its inversion oracle is strictly less than the number of queries it makes to its challenge oracle. (When the adversary makes one challenge query and no inversion queries, this is the standard discrete logarithm assumption, meaning the standard assumption of one-wayness of the discrete exponentiation function.)

The benefits of this result are analogous to those for GQ. Although the assumption is relatively novel and strong, our result reduces the security of the Schnorr identification scheme to a question about the hardness of a number-theoretic problem, thereby freeing a cryptanalyst from consideration of attacks related to the identification problem itself.

1.4 Discussion and related work

Within the large class of FS follow-on identification schemes, proven security properties vary. Some like GQ and Schnorr did not have proofs of security against active or concurrent attacks. However, the FS scheme itself can be proven secure against impersonation under active and concurrent attacks assuming factoring is hard by exploiting its witness-indistinguishability (WI) and proof-of-knowledge (POK) properties. Okamoto's discrete logarithm based scheme [22] is also WI and a POK, and can thus be proven secure against impersonation under active and concurrent attacks, assuming hardness of the discrete logarithm problem. Similar results hold for other schemes having the WI and POK properties. However, GQ and Schnorr are not WI, since there is only one secret key corresponding to a given public key, so these techniques do not work for them. On the other hand, they are preferable in terms of cost. Both have smaller key size than FS, and Schnorr is more efficient than Okamoto.

The so-called 2^m-th root identification scheme can be viewed as the analogue of the GQ scheme with the RSA encryption exponent e replaced by a power of two, or as a special case of the Ong-Schnorr scheme [23]. This has been proven secure against impersonation under active attack assuming factoring is hard [28, 27]. As far as we know, its security against impersonation under concurrent attack is an open question.

The signature schemes obtained from the GQ and Schnorr identification schemes via the Fiat-Shamir transform are already known to be provably-secure in the random oracle model assuming, respectively, the one-wayness of RSA and the hardness of the discrete logarithm problem [25], yet the security of the ID schemes against impersonation under active attack has remained open. This is not a contradiction, since the security of the signature scheme in the random oracle model relies on relatively weak security properties of the ID scheme, namely the security of the latter against impersonation under passive attack [1].

Reset attacks (where the cheating verifier can reset the internal state of prover clones with which it interacts [10, 3]) are not considered here since GQ and Schnorr, as with all proof-of-knowledge based schemes, are insecure against these attacks.

2 Definitions

The empty string is denoted ε. If x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size.

ID SCHEMES. An *identification* (ID) scheme $\mathcal{ID} = (\mathcal{K}, P, V)$ is a triple of randomized algorithms. On input security parameter $k \in \mathbb{N}$, the poly(k)-time key-generation algorithm $\mathcal{K}$ returns a pair consisting of a public key pk and a matching secret key sk. P and V are polynomial-time algorithms that implement the prover and verifier, respectively. We require the natural correctness condition, namely that the boolean decision produced by V, in the interaction in which P has input pk, sk and V has input pk, is one with probability one. This probability is over the coin tosses of both parties. We assume that the first and last moves in the interaction always belong to the prover.

The following security notion uses the basic two-phase framework of [14] in which, in a first phase, the adversary attacks the secret-key equipped P, and then, in a second phase, plays the role of cheating prover, trying to make V accept. We define and prove security only for impersonation under concurrent attack, since the usual (serial) active attack [14] is a special case of a concurrent attack.

IMPERSONATION UNDER CONCURRENT ATTACK. An *imp-ca adversary* $A = (\widehat{V}, \widehat{P})$ is a pair of randomized polynomial-time algorithms, the *cheating verifier* and *cheating prover*, respectively. We consider a game having two phases. In the first phase, $\mathcal{K}$ is run on input k to produce (pk, sk), a random tape is chosen for $\widehat{V}$ and it is given input pk. It then interacts concurrently with different *clones* of prover P, all clones having independent random tapes and being initialized with pk, sk. Specifically, we view P as a function that takes an incoming message and current state and returns an outgoing message and updated state. Cheating verifier $\widehat{V}$ can issue a request of the form (ε, i). As a result, a fresh random tape R_i is chosen, the initial state St_i of clone i is set to (pk, sk, R_i), the operation $(M_{\mathrm{out}}, St_i) \leftarrow P(\varepsilon; St_i)$ is executed, M_{out} is returned to $\widehat{V}$, and the updated St_i is saved as the new state of clone i. Subsequently, $\widehat{V}$ can issue a request of the form (M, i), in which case message M is sent to clone i, who computes $(M_{\text{out}}, St_i) \leftarrow P(M; St_i)$, returns M_{out} to $\widehat{V}$, and saves the updated state St_i. These requests of $\widehat{V}$ can be arbitrarily interleaved. Eventually, $\widehat{V}$ outputs some state information St and stops, ending the first phase. In the second phase of the game, the cheating prover $\widehat{P}$ is initialized with St, verifier V is initialized with pk and freshly chosen coins, and $\widehat{P}$ and V interact. We say that adversary A wins if V accepts in this interaction, and the *imp-ca advantage* of A, denoted

$$\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{ID},A}(k)$$

is the probability that A wins, taken over the coins of $\mathcal{K}$, the coins of $\widehat{V}$, the coins of the prover clones, and the coins of V. (There is no need to give $\widehat{P}$ separate coins, or even pk, since it can get them from $\widehat{V}$ via St.) We say that $\mathcal{ID}$ is *secure against impersonation under concurrent attack* (IMP-CA secure) if the function

$$\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{ID},A}(\cdot)$$

is negligible for all imp-ca adversaries A of time complexity polynomial in the security parameter k.

We adopt the convention that the *time complexity* of imp-ca adversary A does not include the time taken by the prover clones and the verifier to compute replies to the adversary's requests. Rather we view these as oracles, each returning replies in unit time. Barring this, the time complexity of A is the execution time of the entire two-phase game, including the time taken for key generation and initializations. This convention simplifies concrete security considerations.

An active attack [14] is captured by considering cheating verifiers that interact serially, one by one, with prover clones. (This means the cheating verifier initializes a clone and finishes interacting with it before starting up another one.)

Comments. We clarify that we stay within the two-phase framework of [14] even while considering concurrent attacks, in the sense that the first phase (in which the adversary mounts a concurrent attack on the secret-key equipped P) is assumed to be completed before the start of the second phase (in which the adversary plays the role of cheating prover and tries to make V accept). This reflects applications such as smart card based identification for ATMs [14]. For identification over the Internet, it is more suitable to consider adversaries that can interact with the prover or prover clones even while they are interacting with the verifier in an attempt to make the latter accept. With this, one moves into the domain of authenticated key-exchange protocols which is definitionally more complex (see for example [9, 8, 29, 11]) and where identification without an associated exchange of a session-key is of little practical value.

3 Reset lemma

We refer to a three-move protocol of the form depicted in Figure 1 as *canonical*. The prover's first message is called its *commitment*. The verifier selects a *challenge* uniformly at random from a set ChSet_v associated to its input v, and, upon receiving a *response* Rsp from the prover, applies a deterministic *decision*

$$
\begin{array}{rcl}
\underline{\text{Prover}}\ Q & & \underline{\text{Verifier}} \\
\text{Initial State } St = (q, R) & & \\
(\textsc{Cmt}, St) \leftarrow Q(\varepsilon; St) & & \\
 & \xrightarrow{\textsc{Cmt}} & \\
 & & \textsc{Ch} \stackrel{R}{\leftarrow} \mathrm{ChSet}_v \\
 & \xleftarrow{\textsc{Ch}} & \\
(\textsc{Rsp}, St) \leftarrow Q(\textsc{Ch}; St) & & \\
 & \xrightarrow{\textsc{Rsp}} & \\
 & & d \leftarrow \mathrm{DEC}_v(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})
\end{array}
$$

Fig. 1. A canonical protocol. Prover Q has input q and random tape R, and maintains state St. The verifier has input v and returns boolean decision d.

---

predicate $\mathrm{DEC}_v(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ to compute a boolean decision. The verifier is said to be *represented* by the pair $(\mathrm{ChSet}, \mathrm{DEC})$ which, given the verifier input v, defines the challenge set and decision predicate.

A prover is identified with a function Q that given an incoming message M_{in} (this is ε when the prover is initiating the protocol) and its current state St, returns an outgoing message M_{out} and an updated state. The initial state of the prover is (q, R), where q is an input for the prover and R is a random tape.

The following lemma, which we call the Reset Lemma, upper bounds the probability that a (cheating) prover Q can convince the verifier to accept as a function of the probability that a certain experiment based on resetting the prover yields two accepting conversation transcripts. We will use this lemma in our proofs of security of the GQ and the Schnorr schemes at the time of exploiting their proof-of-knowledge properties. In the past such analyses were based on the techniques of [14] who considered certain "execution trees" corresponding to the interaction, and their "heavy nodes." The Reset Lemma provides a slightly better bound, has a simple proof, and is general enough to be applicable in numerous settings, saving the need to apply the techniques of [14] from scratch in each analysis, and may thus be of independent interest. Note that the lemma makes no mention of proofs of knowledge; it is just about relating two probabilities. The formulation and proof of the lemma generalize some analyses in [4].

Lemma 1. (Reset Lemma) *Let Q be a prover in a canonical protocol with a verifier represented by* $(\mathrm{ChSet}, \mathrm{DEC})$*, and let q, v be inputs for the prover and verifier, respectively. Let* $\mathsf{acc}(q, v)$ *be the probability that the verifier accepts in its interaction with Q, namely the probability that the following experiment returns 1:*

Choose random tape R for Q ; $St \leftarrow (q, R)$; $(\textsc{Cmt}, St) \leftarrow Q(\varepsilon; St)$
$\textsc{Ch} \stackrel{R}{\leftarrow} \mathrm{ChSet}_v$; $(\textsc{Rsp}, St) \leftarrow Q(\textsc{Ch}; St)$; $d \leftarrow \mathrm{DEC}_v(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$
Return d

Let $\mathsf{res}(q, v)$ *be the probability that the following reset experiment returns 1:*

> *Choose random tape* R *for* Q ; $St \leftarrow (q, R)$; $(\textsc{Cmt}, St) \leftarrow Q(\varepsilon; St)$
> $\textsc{Ch}_1 \stackrel{R}{\leftarrow} \mathrm{ChSet}_v$; $(\textsc{Rsp}_1, St) \leftarrow Q(\textsc{Ch}_1; St)$; $d_1 \leftarrow \mathrm{DEC}_v(\textsc{Cmt}, \textsc{Ch}_1, \textsc{Rsp}_1)$
> $\textsc{Ch}_2 \stackrel{R}{\leftarrow} \mathrm{ChSet}_v$; $(\textsc{Rsp}_2, St) \leftarrow Q(\textsc{Ch}_2; St)$; $d_2 \leftarrow \mathrm{DEC}_v(\textsc{Cmt}, \textsc{Ch}_2, \textsc{Rsp}_2)$
> *If* ($d_1 = 1$ *AND* $d_2 = 1$ *AND* $\textsc{Ch}_1 \neq \textsc{Ch}_2$) *then return 1 else return 0*

Then

$$\mathsf{acc}(q, v) \;\leq\; \frac{1}{|\mathrm{ChSet}_v|} + \sqrt{\mathsf{res}(q, v)} \; . \quad \blacksquare$$

Proof (Lemma 1). With q, v fixed, let r denote the length of the prover's random tape. For $R \in \{0,1\}^r$ let $\textsc{Cmt}(q, R)$ denote Q's commitment when it has input q and random tape R, and let $\textsc{Rsp}(q, R, \textsc{Ch})$ denote the response provided by Q to verifier challenge $\textsc{Ch} \in \mathrm{ChSet}_v$ when Q has input q and random tape R. We define functions $\mathsf{X}, \mathsf{Y}\colon \{0,1\}^r \to [0,1]$ as follows. For each $R \in \{0,1\}^r$ we let

$$\mathsf{X}(R) \;=\; \Pr\left[\, \mathrm{DEC}_v(\textsc{Cmt}(q, R), \textsc{Ch}, \textsc{Rsp}(q, R, \textsc{Ch})) = 1 \,\right] ,$$

the probability being over a random choice of $\textsc{Ch}$ from ChSet_v. For each $R \in \{0,1\}^r$ we let

$$\mathsf{Y}(R) \;=\; \Pr\left[\begin{array}{l} \mathrm{DEC}_v(\textsc{Cmt}(q, R), \textsc{Ch}_1, \textsc{Rsp}(q, R, \textsc{Ch}_1)) = 1 \text{ and} \\ \mathrm{DEC}_v(\textsc{Cmt}(q, R), \textsc{Ch}_2, \textsc{Rsp}(q, R, \textsc{Ch}_2)) = 1 \text{ and} \\ \textsc{Ch}_1 \neq \textsc{Ch}_2 \end{array}\right] ,$$

the probability being over random and independent choices of $\textsc{Ch}_1$ and $\textsc{Ch}_2$ from ChSet_v. A conditioning argument shows that for any $R \in \{0,1\}^r$

$$\mathsf{Y}(R) \;\geq\; \mathsf{X}(R) \cdot \left[\mathsf{X}(R) - \frac{1}{|\mathrm{ChSet}_v|}\right] .$$

We view X, Y as random variables over the sample space $\{0,1\}^r$ of coins of Q. Then letting $p = 1/|\mathrm{ChSet}_v|$ and using the above we have

$$\begin{aligned}
\mathsf{res}(q, v) &= \mathbf{E}\left[\mathsf{Y}\right] \\
&\geq \mathbf{E}\left[\mathsf{X} \cdot (\mathsf{X} - p)\right] \\
&= \mathbf{E}\left[\mathsf{X}^2\right] - p \cdot \mathbf{E}\left[\mathsf{X}\right] \\
&\geq \mathbf{E}\left[\mathsf{X}\right]^2 - p \cdot \mathbf{E}\left[\mathsf{X}\right] \\
&= \mathsf{acc}(q, v)^2 - p \cdot \mathsf{acc}(q, v) \; .
\end{aligned}$$

In the fourth line above, we used Jensen's inequality[1] applied to the convex function $f(x) = x^2$. Using the above we have

$$\left(\mathsf{acc}(q, v) - \frac{p}{2}\right)^2 \;=\; \mathsf{acc}(q, v)^2 - p \cdot \mathsf{acc}(q, v) + \frac{p^2}{4} \;\leq\; \mathsf{res}(q, v) + \frac{p^2}{4} \; .$$

---

[1] Jensen's inequality states that if f is a convex function and X is a random variable, then $\mathbf{E}\left[\, f(X) \,\right] \geq f(\mathbf{E}\left[X\right])$.

Algorithm $\mathcal{K}(k)$	<u>Prover</u> P		<u>Verifier</u> V
$(N,e,d) \leftarrow \mathcal{K}_{\mathrm{rsa}}(k)$	$y \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$		
$x \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$	$Y \leftarrow y^e \bmod N$		
$X \leftarrow x^e \bmod N$		$\xrightarrow{Y}$	
$pk \leftarrow (N,e,X)$			$c \stackrel{R}{\leftarrow} \{0,1\}^{l(k)}$
$sk \leftarrow (N,x)$		$\xleftarrow{c}$	
Return (pk, sk)	$z \leftarrow yx^c \bmod N$		
		$\xrightarrow{z}$	
			If $z^e \equiv YX^c \pmod N$ then $d \leftarrow 1$ else $d \leftarrow 0$

Fig. 2. GQ identification scheme. Prover P has input $pk = (N,e,X)$ and $sk = (N,x)$. Verifier V has input pk.

Taking the square-root of both sides of the above, and using the fact that $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for all real numbers $a, b \ge 0$, we get

$$\mathsf{acc}(q,v) - \frac{p}{2} \le \sqrt{\mathsf{res}(q,v) + \frac{p^2}{4}} \le \sqrt{\mathsf{res}(q,v)} + \sqrt{\frac{p^2}{4}} = \sqrt{\mathsf{res}(q,v)} + \frac{p}{2} .$$

Re-arranging terms gives us the desired conclusion. ▮

4 Security of GQ under concurrent attack

A randomized, poly(k)-time algorithm $\mathcal{K}_{\mathrm{rsa}}$ is said to be a *prime-exponent RSA key generator* if on input security parameter $k \in \mathbb{N}$, its output is a triple (N, e, d) where N is the product of two distinct primes, $2^{k-1} \le N < 2^k$ (N is k bits long), $e < \varphi(N)$ is an odd prime, $\gcd(d, \varphi(N)) = 1$, and $ed \equiv 1 \pmod{\varphi(N)}$. We do not pin down any specific such generator. Rather it is a parameter of the GQ identification scheme, and security is proved based on an assumption about it.

GQ identification scheme. Let $\mathcal{K}_{\mathrm{rsa}}$ be a prime-exponent RSA key generator and let $l\colon \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable, polynomially-bounded function such that $2^{l(k)} < e$ for any e output by $\mathcal{K}_{\mathrm{rsa}}$ on input k. The *GQ identification scheme* associated to $\mathcal{K}_{\mathrm{rsa}}$ and challenge length l is the ID scheme whose constituent algorithms are depicted in Figure 2. The prover's commitment is a random element $Y \in \mathbb{Z}_N^*$. For any verifier input $pk = (N, e, X)$, $\mathrm{ChSet}_{pk} = \{0,1\}^{l(k)}$. A challenge $c \in \mathrm{ChSet}_{pk}$ is interpreted in the natural way as an integer in the set $\{0, \ldots, 2^{l(k)} - 1\}$ in the ensuing computations. Due to the assumption that $2^{l(k)} < e$, the challenge is in $\mathbb{Z}_e$. The verifier's decision predicate $\mathrm{DEC}_{pk}(Y, c, z)$ evaluates to 1 if and only if z is the RSA-inverse of $YX^c \bmod N$.

RSA assumption. We recall the notion of security under one more inversion (omi) [5]. An *rsa-omi* adversary is a randomized, polynomial-time algorithm I that gets input N, e and has access to two oracles. The first is an RSA-inversion oracle $(\cdot)^d \bmod N$ that given $Y \in \mathbb{Z}_N^*$ returns $Y^d \bmod N$. The second is a challenge oracle that, each time it is invoked (it takes no inputs), returns a random challenge point $W \in \mathbb{Z}_N^*$. The game considered is to run $\mathcal{K}_{\mathrm{rsa}}(k)$ to get N, e, d and then run $I(N, e)$ with its oracles. Let $W_1, \ldots, W_n$ denote the challenges returned by I's challenge oracle. We say that I wins if its output is a sequence of points $w_1, \ldots, w_n \in \mathbb{Z}_N^*$ satisfying $w_i \equiv W_i^d \pmod{N}$ —meaning I inverts all the challenge points— and also the number of queries made by I to its RSA-inversion oracle is *strictly less than* n. The *rsa-omi advantage* of I, denoted $\mathbf{Adv}^{\text{rsa-omi}}_{\mathcal{K}_{\mathrm{rsa}},I}(k)$, is the probability that I wins, taken over the coins of $\mathcal{K}_{\mathrm{rsa}}$, the coins of I, and the coins used by the challenge oracle across its invocations. We say that $\mathcal{K}_{\mathrm{rsa}}$ is OMI secure if the function $\mathbf{Adv}^{\text{rsa-omi}}_{\mathcal{K}_{\mathrm{rsa}},I}(\cdot)$ is negligible for any rsa-omi adversary I of time complexity polynomial in k.

We adopt the convention that the *time complexity* of an rsa-omi adversary I is the execution time of the entire game, including the time taken for key generation and one time unit for each reply to an oracle query. (The time taken by the oracles to compute replies to the adversary's queries is not included.)

Result. The following theorem shows that the advantage of any imp-ca attacker against the GQ scheme can be upper bounded via the advantage of a related rsa-omi adversary and a function of the challenge length. The theorem shows the concrete security of the reduction.

Theorem 1. *Let $\mathcal{ID} = (\mathcal{K}, P, V)$ be the GQ identification scheme associated to prime-exponent RSA key generator $\mathcal{K}_{\mathrm{rsa}}$ and challenge length l. Let $A = (\widehat{V}, \widehat{P})$ be an imp-ca adversary of time complexity $t(\cdot)$ attacking $\mathcal{ID}$. Then there exists an rsa-omi adversary I attacking $\mathcal{K}_{\mathrm{rsa}}$ such that for every k*

$$\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{ID},A}(k) \;\leq\; 2^{-l(k)} + \sqrt{\mathbf{Adv}^{\text{rsa-omi}}_{\mathcal{K}_{\mathrm{rsa}},I}(k)}\,. \tag{1}$$

Furthermore, the time complexity of I is $2t(k) + O(k^4 + (n(k)+1) \cdot l(k) \cdot k^2)$, where $n(k)$ is the number of prover clones with which $\widehat{V}$ interacts.

Based on this theorem, which we will prove later, we can easily provide the following security result for the GQ scheme. In this result, we assume that the challenge length l is super-logarithmic in the security parameter, which means that $2^{-l(\cdot)}$ is negligible. This assumption is necessary, since otherwise the GQ scheme can be broken merely by guessing the verifier's challenge.

Corollary 1. *If prime-exponent RSA key generator $\mathcal{K}_{\mathrm{rsa}}$ is OMI secure and challenge length l satisfies $l(k) = \omega(\log(k))$, then the GQ identification scheme associated to $\mathcal{K}_{\mathrm{rsa}}$ and l is secure against impersonation under both active and concurrent attacks.*

We proceed to prove Theorem 1.

Proof (Theorem 1). We assume wlog that $\widehat{V}$ never repeats a request. Fix $k \in \mathbb{N}$ and let (N, e, d) be an output of $\mathcal{K}_{\mathrm{rsa}}$ running on input k. Adversary I has access to an RSA-inversion oracle $(\cdot)^d \bmod N$ and a challenge oracle $\mathcal{O}_N$ that takes no inputs and returns a random challenge point $W \in \mathbb{Z}_N^*$ each time it is invoked.

Adversary $I^{(\cdot)^d \bmod N, \mathcal{O}_N}(N, e)$
  Make a query to $\mathcal{O}_N$ and let W_0 be the response ; $pk \leftarrow (N, e, W_0)$
  Choose a random tape R for $\widehat{V}$; Initialize $\widehat{V}$ with (pk, R) ; $n \leftarrow 0$
  Run $\widehat{V}$ answering its requests as follows:
    When $\widehat{V}$ issues a request of the form (ε, i) do
      $n \leftarrow n + 1$; Make a query to $\mathcal{O}_N$, let W_i be the response
      and return W_i to $\widehat{V}$
    When $\widehat{V}$ issues a request of the form (c, i), where $c \in \{0,1\}^{l(k)}$, do
      $c_i \leftarrow c$; Make query $W_i W_0^{c_i}$ to $(\cdot)^d \bmod N$, let z_i be the response
      and return z_i to $\widehat{V}$
  Until $\widehat{V}$ outputs state information St and stops
  $St \leftarrow (St, \varepsilon)$; $(Y, St) \leftarrow \widehat{P}(\varepsilon; St)$
  $\text{CH}_1 \stackrel{R}{\leftarrow} \{0,1\}^{l(k)}$; $(\text{RSP}_1, \overline{St}) \leftarrow \widehat{P}(\text{CH}_1; St)$
  If $\text{RSP}_1^e \equiv Y W_0^{\text{CH}_1} \pmod N$ then $d_1 \leftarrow 1$ else $d_1 \leftarrow 0$
  $\text{CH}_2 \stackrel{R}{\leftarrow} \{0,1\}^{l(k)}$; $(\text{RSP}_2, \overline{St}) \leftarrow \widehat{P}(\text{CH}_2; St)$
  If $\text{RSP}_2^e \equiv Y W_0^{\text{CH}_2} \pmod N$ then $d_2 \leftarrow 1$ else $d_2 \leftarrow 0$
  If $(d_1 = 1 \text{ AND } d_2 = 1 \text{ AND } \text{CH}_1 \neq \text{CH}_2)$ then
    $z \leftarrow \text{RSP}_1 \cdot \text{RSP}_2^{-1} \bmod N$; $(\overline{d}, a, b) \leftarrow \text{EGCD}(e, \text{CH}_1 - \text{CH}_2)$
    $w_0 \leftarrow W_0^a z^b \bmod N$; For $i = 1$ to n do $w_i \leftarrow z_i w_0^{-c_i} \bmod N$
    Return $w_0, w_1, \ldots, w_n$
  else Return $\perp$

Fig. 3. rsa-omi adversary I for the proof of Theorem 1.

The adversary's goal is to invert all the challenges returned by $\mathcal{O}_N$, while making fewer queries to its RSA-inversion oracle then the number of such challenges.

A detailed description of the adversary is in Figure 3. It queries its challenge oracle to obtain a random element $W_0 \in \mathbb{Z}_N^*$ and uses it to create a public key pk for adversary A. It then uses A to achieve its goal by running $\widehat{V}$ and playing the role of the prover clones to answer its requests. In response to a request (ε, i), I queries its challenge oracle and returns the answer W_i to $\widehat{V}$. By the definition of prover P, from $\widehat{V}$'s perspective, this is equivalent to picking a random tape R_i for prover clone i, initializing clone i with state pk, R_i, computing clone i's commitment W_i, and returning the commitment to $\widehat{V}$. I is not in possession of the secret key $sk = (N, W_0^d \bmod N)$ corresponding to pk, which the prover clones would use to respond to $\widehat{V}$'s requests of the form (c, i), where $c \in \{0,1\}^{l(k)}$, but it compensates using its access to the RSA-inversion oracle to answer these requests. Specifically, in response to request (c, i), I makes the query $W_i W_0^c$ to its inversion oracle and returns the answer z_i to $\widehat{V}$. Since $z_i = (W_i W_0^c)^d \bmod N = W_i^d (W_0^d)^c \bmod N$, this is exactly the response that clone i would return to $\widehat{V}$. Hence I simulates the behavior of the prover clones perfectly.

If $n(k)$ is the number of prover clones with which $\widehat{V}$ interacts, when $\widehat{V}$ stops I has made $n(k)$ queries to its RSA-inversion oracle and it needs to invert $n(k)+1$ points. I attempts to extract from $\widehat{P}$, initialized with the output of $\widehat{V}$, the RSA-inverse of challenge W_0. To do so, it runs $\widehat{P}$ obtaining its commitment, selects two independent random challenges from $\{0,1\}^{l(k)}$, runs $\widehat{P}$ to obtain its response to each of these challenges (with the same state), and evaluates the verifier's decision predicate on $\widehat{P}$'s commitment and each challenge-response pair. If the decision predicate evaluates to 1, meaning $\widehat{P}$ makes the verifier accept, on both accounts and the challenges are different, then I extracts the inverse of W_0 as follows. It computes the quotient mod N of the cheating prover's responses to the challenges and sets z to this value. We observe that $z^e \equiv W_0^{\mathrm{CH}_1-\mathrm{CH}_2} \pmod N$. Then I uses the extended Euclid algorithm, EGCD, to compute $(\overline{d}, a, b)$, where $\overline{d} = \gcd(e, \mathrm{CH}_1 - \mathrm{CH}_2)$ and $a, b \in \mathbb{Z}$ are such that $ae + b(\mathrm{CH}_1 - \mathrm{CH}_2) = \overline{d}$. By the assumptions that e is prime and $2^{l(k)} < e$ (which implies $\mathrm{CH}_1, \mathrm{CH}_2 \in \mathbb{Z}_e$), $\overline{d} = 1$. Hence $ae + b(\mathrm{CH}_1 - \mathrm{CH}_2) = 1$. Therefore, modulo N we have

$$W_0 \;\equiv\; W_0^{ae} W_0^{b(\mathrm{CH}_1-\mathrm{CH}_2)} \;\equiv\; W_0^{ae}(W_0^{\mathrm{CH}_1-\mathrm{CH}_2})^b \;\equiv\; W_0^{ae}(z^e)^b \;\equiv\; (W_0^a z^b)^e \,.$$

This shows that $w_0 = W_0^a z^b \bmod N$ is the RSA-inverse of W_0. For $i = 1, \ldots, n(k)$, I computes the inverse of the i-th challenge point as $w_i = z_i w_0^{-c_i} \bmod N$. To prove that this computation yields the desired RSA-inverse, we show that $w_i^e \equiv W_i \pmod N$. Since z_i is the inverse of $W_i W_0^{c_i}$ and w_0 is the inverse of W_0,

$$w_i^e \;\equiv\; (z_i w_0^{-c_i})^e \;\equiv\; z_i^e (w_0^e)^{-c_i} \;\equiv\; W_i W_0^{c_i} W_0^{-c_i} \;\equiv\; W_i \pmod N \,.$$

If the decision predicate does not evaluate to 1 on both occasions or the challenges coincide, then I fails. Therefore, I wins if and only if $d_1 = 1$, $d_2 = 1$ and $\mathrm{CH}_1 \neq \mathrm{CH}_2$. We proceed to relate the probability of this event with the imp-ca advantage of adversary A.

We observe that pk has the same distribution as in the two-phase game that defines a concurrent attack. Since I simulates the environment provided to $\widehat{V}$ in that game perfectly, $\widehat{V}$ behaves as it does when performing a concurrent attack against $\mathcal{ID}$, and $\widehat{P}$ is given state information with the same distribution as in that case. Therefore, the probability that $d_1 = 1$ is exactly $\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{ID},A}(k)$.

Let $\mathsf{acc}(St, pk)$ denote the probability that $d_1 = 1$ when the public key created by I is pk and the output of $\widehat{V}$ is St. (This probability is over the choice of challenge CH_1.) Let $\mathsf{res}(St, pk)$ denote the probability that $d_1 = 1$, $d_2 = 1$ and $\mathrm{CH}_1 \neq \mathrm{CH}_2$ when the public key created by I is pk and the output of $\widehat{V}$ is St. (The probability here is over the choice of challenges CH_1 and CH_2.) Then, if $\mathbf{E}[\,\cdot\,]$ denotes the expectation of random variable $\cdot$ over the choice of pk and St, the probability that $d_1 = 1$ is $\mathbf{E}[\,\mathsf{acc}(St, pk)\,]$, and the probability that I wins is $\mathbf{E}[\,\mathsf{res}(St, pk)\,]$. Applying the Reset Lemma to $\widehat{P}$ with input St and verifier V with input pk, where the latter is implemented by I, we have

$$\mathsf{acc}(St, pk) \;\leq\; 2^{-l(k)} + \sqrt{\mathsf{res}(St, pk)} \,.$$

We obtain Equation (1) as follows.

$$\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{ID},A}(k) = \mathbf{E}[\,\mathsf{acc}(St, pk)\,]$$

Algorithm $\mathcal{K}(k)$	Prover P		Verifier V
$(q, g) \leftarrow \mathcal{K}_{\text{dl}}(k)$	$y \stackrel{R}{\leftarrow} \mathbb{Z}_q$		
$x \stackrel{R}{\leftarrow} \mathbb{Z}_q$	$Y \leftarrow g^y$		
$X \leftarrow g^x$		$\xrightarrow{Y}$	
$pk \leftarrow (q, g, X)$			$c \stackrel{R}{\leftarrow} \{0, 1\}^{l(k)}$
$sk \leftarrow (q, x)$		$\xleftarrow{c}$	
Return (pk, sk)	$z \leftarrow y + cx \bmod q$		
		$\xrightarrow{z}$	
			If $g^z = YX^c$ then $d \leftarrow 1$ else $d \leftarrow 0$

Fig. 4. Schnorr identification scheme. Prover P has input $pk = (q, g, X)$ and $sk = (q, x)$. Verifier V has input pk.

$$
\begin{aligned}
&\leq \mathbf{E}\left[\, 2^{-l(k)} + \sqrt{\mathsf{res}(St, pk)}\,\right] \\
&= 2^{-l(k)} + \mathbf{E}\left[\sqrt{\mathsf{res}(St, pk)}\right] \\
&\leq 2^{-l(k)} + \sqrt{\mathbf{E}\,[\,\mathsf{res}(St, pk)\,]} \\
&= 2^{-l(k)} + \sqrt{\mathbf{Adv}^{\text{rsa-omi}}_{\mathcal{K}_{\text{rsa}}, I}(k)}\ ,
\end{aligned}
$$

where the last inequality follows from Jensen's inequality[2] applied to the concave function $f(x) = \sqrt{x}$.

The proof of the claim about the time complexity of adversary I is in the full version of this paper [7]. ▮

5 Security of Schnorr under concurrent attack

A randomized, poly(k)-time algorithm $\mathcal{K}_{\text{dl}}$ is said to be a *discrete logarithm parameter generator* if given security parameter $k \in \mathbb{N}$, it outputs a pair (q, g) where q is a prime such that $q \,|\, p - 1$ for a prime p with $2^{k-1} \leq p < 2^k$ (p is k bits long), and g is a generator of G_q, a subgroup of $\mathbb{Z}_p^*$ of order q. As before, we do not pin down any specific such generator. The generator is a parameter of the Schnorr scheme, and security is proved based on an assumption about it.

Schnorr identification scheme. Let $\mathcal{K}_{\text{dl}}$ be a discrete logarithm parameter generator and let $l\colon \mathbb{N} \to \mathbb{N}$ be a polynomial-time computable, polynomially-bounded function such that $2^{l(k)} < q$ for any q output by $\mathcal{K}_{\text{dl}}$ on input k. The *Schnorr identification scheme* associated to $\mathcal{K}_{\text{dl}}$ and challenge length l is the

[2] Jensen's inequality states that if f is a concave function and X is a random variable, then $\mathbf{E}\,[\,f(X)\,] \leq f(\mathbf{E}\,[X])$.

ID scheme whose constituent algorithms are depicted in Figure 4. The prover's commitment is a random element $Y \in G_q$. For any verifier input $pk = (q, g, X)$, $\mathrm{ChSet}_{pk} = \{0,1\}^{l(k)}$. A challenge $c \in \mathrm{ChSet}_{pk}$ is interpreted as an integer in the set $\{0, \ldots, 2^{l(k)} - 1\}$ in the ensuing computations. The assumption that $2^{l(k)} < q$ implies that the challenge is in $\mathbb{Z}_q$. The verifier's decision predicate $\mathrm{DEC}_{pk}(Y, c, z)$ evaluates to 1 if and only if z is the discrete log of YX^c.

DL assumption. We recall the notion of security under one more discrete logarithm (omdl) [5]. An *omdl* adversary is a randomized, polynomial-time algorithm I that gets input q, g and has access to two oracles. The first is a discrete log oracle $\mathrm{DLog}_{G_q,g}(\cdot)$ that given $Y \in G_q$ returns $y \in \mathbb{Z}_q$ such that $g^y = Y$. The second is a challenge oracle that, each time it is invoked, returns a random challenge point $W \in G_q$. The game considered is to run $\mathcal{K}_{\mathrm{dl}}(k)$ to get q, g and then run $I(q, g)$ with its oracles. Let $W_1, \ldots, W_n$ denote the challenges returned by I's challenge oracle. We say that I wins if its output is a sequence of points $w_1, \ldots, w_n \in Z_q$ satisfying $g^{w_i} = W_i$, and also the number of queries made by I to its discrete log oracle is *strictly less than* n. The *omdl advantage* of I, denoted $\mathbf{Adv}^{\mathrm{omdl}}_{\mathcal{K}_{\mathrm{dl}},I}(k)$, is the probability that I wins, taken over the coins of $\mathcal{K}_{\mathrm{dl}}$, the coins of I, and the coins used by the challenge oracle across its invocations. We say that $\mathcal{K}_{\mathrm{dl}}$ is OMDL secure if the function $\mathbf{Adv}^{\mathrm{omdl}}_{\mathcal{K}_{\mathrm{dl}},I}(\cdot)$ is negligible for any omdl adversary I of time complexity polynomial in k.

We adopt the same convention regarding time complexity as in the case of an rsa-omi adversary.

Result. The following theorem guarantees that the advantage of any imp-ca adversary attacking the Schnorr scheme can be upper bounded via the advantage of a related omdl adversary and a function of the challenge length.

Theorem 2. *Let $\mathcal{ID} = (\mathcal{K}, P, V)$ be the Schnorr identification scheme associated to discrete logarithm parameter generator $\mathcal{K}_{\mathrm{dl}}$ and challenge length l. Let $A = (\widehat{V}, \widehat{P})$ be an imp-ca adversary of time complexity $t(\cdot)$ attacking $\mathcal{ID}$. Then there exists an omdl adversary I attacking $\mathcal{K}_{\mathrm{dl}}$ such that for every k*

$$\mathbf{Adv}^{\mathrm{imp\text{-}ca}}_{\mathcal{ID},A}(k) \;\le\; 2^{-l(k)} + \sqrt{\mathbf{Adv}^{\mathrm{omdl}}_{\mathcal{K}_{\mathrm{dl}},I}(k)}\;. \tag{2}$$

Furthermore, the time complexity of I is $2t(k) + O(k^3 + (l(k) + n(k)) \cdot k^2)$, where $n(k)$ is the number of prover clones with which $\widehat{V}$ interacts.

The proof of Theorem 2 is in the full version of this paper [7]. This theorem implies the following security result for the Schnorr scheme.

Corollary 2. *If discrete logarithm parameter generator $\mathcal{K}_{\mathrm{dl}}$ is OMDL secure and challenge length l satisfies $l(k) = \omega(\log(k))$, then the Schnorr identification scheme associated to $\mathcal{K}_{\mathrm{dl}}$ and l is secure against impersonation under both active and concurrent attacks.*

As in the case of the GQ scheme, the assumption that the challenge length l is super-logarithmic in the security parameter is necessary since otherwise the Schnorr scheme can be broken by guessing the verifier's challenge.

Acknowledgments

The authors were supported in part by NSF grant CCR-0098123, NSF grant ANR-0129617 and an IBM Faculty Partnership Development Award.

References

1. M. Abdalla, J. An, M. Bellare and C. Namprempre. From identification to signatures via the Fiat-Shamir Transform: Minimizing assumptions for security and forward-security. *Advances in Cryptology – EUROCRYPT '02*, Lecture Notes in Computer Science Vol. 2332 , L. Knudsen ed., Springer-Verlag, 2002.
2. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. *Advances in Cryptology – EUROCRYPT '97*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.
3. M. Bellare, M. Fischlin, S. Goldwasser and S. Micali. Identification protocols secure against reset attacks. *Advances in Cryptology – EUROCRYPT '01*, Lecture Notes in Computer Science Vol. 2045, B. Pfitzmann ed., Springer-Verlag, 2001.
4. M. Bellare and S. Miner. A forward-secure digital signature scheme. *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.
5. M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko. The one-more-RSA inversion problems and the security of Chaum's blind signature scheme. Available as *IACR eprint archive Report 2001/002*, `http://eprint.iacr.org/2001/002/`. Preliminary version, entitled "The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme," in *Financial Cryptography '01*, Lecture Notes in Computer Science Vol. 2339, P. Syverson ed., Springer-Verlag, 2001.
6. M. Bellare and G. Neven. Transitive signatures based on factoring and RSA. Manuscript, May 2002.
7. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. Full version of this paper, available via `http://www-cse.ucsd.edu/uers/mihir`.
8. M. Bellare, D. Pointcheval and P. Rogaway. Authenticated key exchange secure against dictionary attacks. *Advances in Cryptology – EUROCRYPT '00*, Lecture Notes in Computer Science Vol. 1807, B. Preneel ed., Springer-Verlag, 2000.
9. 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.
10. 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.
11. R. Canetti and H. Krawczyk. Universally composable notions of key-exchange and secure channels. *Advances in Cryptology – EUROCRYPT '02*, Lecture Notes in Computer Science Vol. 2332 , L. Knudsen ed., Springer-Verlag, 2002.
12. D. Chaum. Blind signatures for untraceable payments. *Advances in Cryptology – CRYPTO '82*, Lecture Notes in Computer Science, Plenum Press, New York and London, 1983, Aug. 1982.

13. R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. In *5th ACM Conference on Computer and Communications Security*, pages 46–51, Singapore, Nov. 1999. ACM Press.
14. U. Feige, A. Fiat and A. Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.
15. U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.
16. 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.
17. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. *Advances in Cryptology – CRYPTO '97*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.
18. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science Vol. 1592, J. Stern ed., Springer-Verlag, 1999.
19. S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, 18(1):186–208, February 1989.
20. L. Guillou and J. J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. *Advances in Cryptology – CRYPTO '88*, Lecture Notes in Computer Science Vol. 403, S. Goldwasser ed., Springer-Verlag, 1988.
21. S. Micali and R. Rivest. Transitive signature schemes. *Topics in Cryptology – CT-RSA '02*, Lecture Notes in Computer Science Vol. 2271 , B. Preneel ed., Springer-Verlag, 2002.
22. 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.
23. H. Ong and C. P. Schnorr. Fast signature generation with a Fiat Shamir–like scheme. *Advances in Cryptology – EUROCRYPT '90*, Lecture Notes in Computer Science Vol. 473, I. Damgård ed., Springer-Verlag, 1990.
24. D. Pointcheval. New public key cryptosystems based on the dependent-RSA problems. *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science Vol. 1592, J. Stern ed., Springer-Verlag, 1999.
25. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
26. C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
27. C. P. Schnorr. Security of the 2^t-root identification and signatures. *Advances in Cryptology – CRYPTO '96*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.
28. V. Shoup. On the security of a practical identification scheme. *Journal of Cryptology*, 12:247–260, 1999.
29. V. Shoup. On formal models for secure key exchange (version 4). *IACR eprint archive Report 1999/012*, `http://eprint.iacr.org/1999/012/`.